Departamento de Informática

PROYECTO FIN DE CARRERA

MODERNIZACIÓN DEL SISTEMA DATAWAREHOUSE EN EMPRESAS DEL SECTOR DE DISTRIBUCIÓN ALIMENTARIA

Autor: Unai Fernández Rivas

Tutor: María Dolores Cuadra Fernández

Leganés, 22 Octubre de 2015

Título: Modernización del Sistema de Data Warehouse en empresas del sector de distribución alimentaria

Autor: Unai Fernández Rivas

Director: María Dolores Cuadra Fernández

EL TRIBUNAL

Presidente: FRANCISCO JAVIER CALLE

Vocal: ANA IGLESIAS

Secretario: ALEJANDRO CALDERÓN

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 22 de Octubre de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

En primero lugar, me gustaría agradecer el esfuerzo que han realizado mis padres para que yo pudiese estudiar lo que me gustaba así como la paciencia y la comprensión que han tenido conmigo, espero que puedan sentirse orgullosos de su hijo.

A nivel profesional, tengo que agradecer a todos mis compañeros, ya que hace hoy más de 7 años me acogieron como un asustado universitario desconocedor del mundo profesional y me facilitaron los medios para poder alcanzar los conocimientos que hoy en día dispongo. Todos y cada uno de ellos forman un gran equipo profesional y humano.

En particular tengo que agradecer a Juan Carlos Morán todo lo que me ha enseñado durante estos años, la cantidad de manuales que me ha mandado leer, los viajes que hemos realizado juntos, las buenos y malos momentos que hemos pasado y por supuesto los maravillosos miércoles de cine.

Tengo que agradecer también a mis amigos la paciencia que han tenido conmigo en los momentos finales del proyecto así como la ayuda que muchos de ellos me han brindado.

No puedo dejar de agradecer la posibilidad que mi tutora me ha brindado para poder realizar este tipo de proyecto, y más con los problemas de compatibilidad laboral que he tenido, mil gracias por la paciencia que has tenido.

Por último, tengo que agradecer enormemente a mi futura esposa todo el apoyo que me ha dado durante este trayecto que hemos realizado conjuntamente, sin ti no lo habría conseguido, eres maravillosa, te quiero mucho.

Unai

Resumen

Modernización del sistema Data Warehouse en empresa del sector de distribución alimentaria es un proyecto que pretender acercar el mundo laboral al universitario. En él se muestra un problema real de una empresa del sector, y se desarrolla la solución elegida por el cliente de forma que software y hardware profesional se puedan conocer en los ámbitos universitarios. Presenta también una visión teórica del concepto "Big Data" y una visión real de la implantación en el sector.

El proyecto acerca el mundo del dato, su almacenamiento, tratamiento y explotación de forma sencilla y presentando las tecnologías punteras del mercado. El proyecto engloba una solución hardware y software, así como un caso de ejemplo de transformación de datos.

Palabras Clave: Data Warehouse, Dato, sector distribución alimentaria, IBM Pure Data for Analytics, IBM InfoSphere Information Server, BIG DATA.

Abstract

Data Warehouse system modernization in the food distribution business sector is a project that try to bring the world of work at the university.

In the project a real problem for enterprises in the sector is shown, the solution chosen by the customer is developed, so that the professional software and hardware can be found on the university environment.

It also presents a theoretical overview of the concept "Big Data" and a real insight into the implementation in the sector.

The project brings the world of data, storage, processing and use of simple and introducing high technologies in the market. The project encompasses a hardware and software solution as well as a case example of data transformation.

Keywords: Data Warehouse, Data, food retail industry, IBM Pure Data for Analytics, IBM InfoSphere Information Server, BIG DATA.

Índice de contenidos

1 INTRODUCCIÓN 12
1.1 MOTIVACIÓN 12
1.2 OBJETIVOS 13
1.3 FASES DEL PROYECTO 14
1.4 HERRAMIENTAS EMPLEADAS 15
1.5 ESQUEMA DE LA MEMORIA 17
2 ESTADO DEL ARTE 18
2.1 ¿QUÉ ES BIG DATA? 18
2.1.1 BIG DATA COMO CONCEPTO 18
2.1.2 BIG DATA EN EL ENTORNO PROFESIONAL ESPAÑOL 23
2.2 SISTEMAS DATA WAREHOUSE 25
2.2.1 Principales soluciones Data Warehouse en el mercado profesional 29
2.2.2 IBM PURE DATA SYSTEM FOR ANALYTICS 29
2.2.3 ORACLE EXADATA x5 31
2.2.4 TERADATA DATA WAREHOUSE 34
2.3 HERRAMIENTAS EXTRACCIÓN, TRANSFORMACIÓN Y CARGA (ETL) 36
2.3.1 Principales herramientas ETL disponibles en el mercado profesional 41
2.3.2 INFORMATICA POWERCENTER 41
2.3.3 IBM INFOSPHERE DATASTAGE 43
2.3.4 ORACLE WAREHOUSE BUILDER 45
2.4 ANALISIS Y PROPUESTA DE SOLUCION DE PROBLEMÁTICA EN UN CLIENTE DEL SECTOR DISTRIBUCIÓN ALIMENTARIA 48
3 SOLUCIÓN DATAWAREHOUSE + ETL, EN CLIENTE DEL SECTOR DISTRIBUCION ALIMENTARIA 54
3.1 ARQUITECTURA DATAWAREHOUSE 54
3.1.1 HARDWARE 55
3.1.2 SOFTWARE 62
3.2 ARQUITECTURA ETL 69
3.2.1 HARDWARE 70
3.2.2 SOFTWARE 75

3.3 CASO DE USO: TRANSFORMACION DE PROGRAMAS RPG+COBOL EN FLUJOS DE TRABAJO DE LA ETL 81

3.3.1 Fase 1: El dato viaja de la tienda al servidor central de recepción 82

3.3.2 Fase 2: Integración de los ficheros de datos en tablas de Netezza 84

3.3.3 Fase 3: Transformación de los datos aplicando la lógica de negocio 99

4 CONCLUSIONES 123

4.1 Conclusiones personales 125

4.2 Trabajos futuros 125

5 PRESUPUESTO 127

5.1 Introducción 127

5.2 Fases del proyecto 127

5.3 Costes 130

5.3.1 Coste de personal 130

5.3.2 Coste de material y herramientas 130

6 GLOSARIO 132

7 REFERENCIAS 134

Índice de ilustraciones

Ilustración 1 - Resultados búsqueda "Big Data" 12
Ilustración 2 - IBM Pure Data System For Analytics 29
Ilustración 3 - Oracle Exadata x5 31
Ilustración 4 - Teradata Data Warehouse Appliance 34
Ilustración 5 - Flujo de datos (ETL) 37
Ilustración 6 - Arquitectura Informatica PowerCenter [Juan Garcés 2013] 42
Ilustración 7 - Arquitectura IBM InfoSphere DataStage 44
Ilustración 8 - Oracle Warehouse Builder components [Oracle 2015] 47
Ilustración 9 - PureData System for Analytics 55
Ilustración 10 - Estrategia de protección de discos 56
Ilustración 11 - Mapa de Red del sistema 57
Ilustración 12 - Esquema Rack PureData 58
Ilustración 13 - BladeCenter + DBA Accelerator 59
Ilustración 14 - Netezza DB Accelerator 60
Ilustración 15 - Comportamiento del sistema ante la caída de un S-Blade 61
Ilustración 16 - Sistema de Procesamiento Paralelo Masivo de Netezza (I) 62
Ilustración 17 - Sistema de Procesamiento Paralelo Masivo de Netezza (II) 63
Ilustración 18 - Proceso de ejecución de un query en Netezza 64
Ilustración 19 - Uso de ZoneMaps en Netezza 66
Ilustración 20 - Familia InfoSphere Information Server 69
Ilustración 21 - 8286-42A IBM Power S824 70
Ilustración 22 - IBM v7000 71
Ilustración 23 - Estructura del Logical Volume Manager [Davis Mendoza Paco] 72
Ilustración 24 - Resumen de configuración de VolumeGroup 73
Ilustración 25 - Resumen de logical volumes del volume group datavg 74
Ilustración 26 - Resumen de los FileSystems creados 74
Ilustración 27 - Detalle de configuración de red (DataStage) 75
Ilustración 28 - Estructura de DataStage 76
Ilustración 29 - Capa Cliente DataStage 77
Ilustración 30 - Capa Servicios DataStage 77
Ilustración 31 - Capa motor DastaStage 78
Ilustración 32 - Capa de Repositorio de Metadatos 80
Ilustración 33 - Flujo de generación da información desde las tiendas 83
Ilustración 34 - Generación DDL iSeries (I) 84
Ilustración 35 - Generación DDL iSeries (II) 85
Ilustración 36 - Generación DDL iSeries (III) 86
Ilustración 37 - Generación DDL iSeries (IV) 86
Ilustración 38 - Generación DDL iSeries (V) 87

Ilustración 39 - Generación DDL iSeries (VI) 87
Ilustración 40 - Generación Tablas Netezza (I) 88
Ilustración 41 - Generación Tablas Netezza (II) 89
Ilustración 42 - DataStage Job: Carga de tablas temporales (TRF) 90
Ilustración 43 - DataStage Job: LIMPIEZA (CARGATRF) 90
Ilustración 44 - DataStage Job: LIMPIAR_TRF (LIMPIEZA) 91
Ilustración 45 - DataStage Job: LIMPIAR_TXT (LIMPIEZA) 91
Ilustración 46 - DataStage Job: GENERAR_LISTADO (CARGATRF) 92
Ilustración 47 - DataStage Job: COMPROBAR_TRF (CARGATRF) 92
Ilustración 48 - DataStage Job: BUCLE (CARGATRF) 93
Ilustración 49 - DataStage Jobs: CARGA_DAT (CARGATRF) 94
Ilustración 50 - DataStage Job: Listado Trabajos CARGADAT 95
Ilustración 51 - DataStage Job: CSATDRFF (CARGADAT) 95
Ilustración 52 - DataStage Job: AUTLINA (CARGADAT) 96
Ilustración 53 - DataStage Job: Detalle Transformer (AUTLINA) 96
Ilustración 54 - DataStage Job: BACKUPDAT (CARGATRF) 97
Ilustración 55 - DataStage Job: BACKUPDAT Ejemplo fichero (CARGATRF) 98
Ilustración 56 - Secuencia Totales Tickets 106
Ilustración 57 - Etapas de control de errores (Totales Tickets) 106
Ilustración 58 - Primera parte diseño (Totales Tickets) 107
Ilustración 59 - Detalle conexión Netezza (Totales Tickets) 107
Ilustración 60 - Definición de columnas, datos de origen (Totales Tickets) 108
Ilustración 61 - Transfomer mapeo de campos (Totales Tickets) 108
Ilustración 62 - Detalle transformaciones (Totales Tickets) 109
Ilustración 63 - Lookup GENTDATF (Movimiento Tickets) 110
Ilustración 64 - Segunda parte diseño (Totales Tickets) 111
Ilustración 65 - Except Join (Totales Tickets) 112
Ilustración 66 - Lookup CALENDAR I (Totales Tickets) 113
Ilustración 67 - Lookup CALENDAR II (Totales Tickets) 114
Ilustración 68 - Lookup TIENDAS (Totales Tickets) 115
Ilustración 69 - Detalle Transformer CODCLI (Totales Tickets) 115
Ilustración 70 - Detalle Lookup TIENDAS I (Totales Tickets) 116
Ilustración 71 - Detalle Lookup TIENDAS II (Totales Tickets) 116
Ilustración 72 - Lookup SOCIOS (Totales Tickets) 118
Ilustración 73 - Lookup01 SOCIOS (Totales Tickets) 119
Ilustración 74 - Lookup02 SOCIOS (Totales Tickets) 119
Ilustración 75 - Lookup03 SOCIOS (Totales Tickets) 120
Ilustración 76 - Transformer desglose NIF SOCIOS (Totales Tickets) 121
Ilustración 77 - Lookup04 SOCIOS (Totales Tickets) 121
Ilustración 78 - Escritura de Datos en tablas destino (Totales Tickets) 122
Ilustración 79 - Diagrama Gantt del proyecto 129

Índice de tablas

Tabla 1 - Características Exadata x5 33
Tabla 2 - Concurrencia de usuarios sistema IBM 52
Tabla 3- Concurrencia de usuarios sistema TERADATA 52
Tabla 4 - Configuración del Sistema 8286-42A 71
Tabla 5 - Datos origen (Movimientos Ticket) 100
Tabla 6 - Datos final (Movimientos Ticket) 101
Tabla 7 - Transformaciones (Movimientos Ticket) 103
Tabla 8 - Etapa LOOKUP, DataStage 110
Tabla 9 - Recursos y tiempo empleado, fase análisis 127
Tabla 10 - Recursos y tiempo empleado, fase desarrollo 128
Tabla 11 - Recursos y tiempo empleado, fase documentación 129
Tabla 12 - Costes de personal para la realización del proyecto 130
Tabla 13 - Costes de material y herramienta para la realización del proyecto 130

1 INTRODUCCIÓN

1.1 MOTIVACIÓN

En la actualidad nos encontramos ante una revolución tecnológica centrada en el "dato" y en el uso que podemos realizar con ellos. Se dice que "Los datos masivos (Big Data) son el nuevo oro" [Viktor Mayer-Schönberger, Kenneth Cukier 2013]

Si realizamos una búsqueda en google con las palabras "Big Data" obtenemos 785 millones de resultados

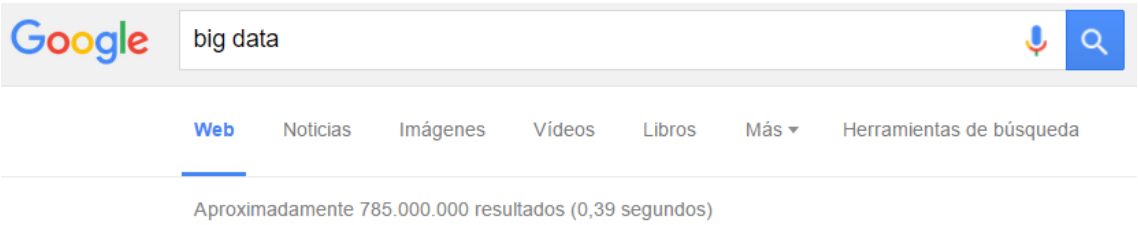

Ilustración 1 - Resultados búsqueda "Big Data"

Nos podemos hacer una idea de la magnitud de información que se aporta sobre este fenómeno.

Este flujo de información provoca que las empresas se planteen realizar mejoras sobre sus sistemas de información actuales para adecuarse a estas tecnologías.

El proyecto tiene como misión resolver un problema real tecnológico al que se enfrenta una empresa que dispone de un almacén de datos obsoleto y con problemas de rendimiento y se plantea resolver esta situación utilizando tecnologías big data para ellos.

Como expondremos a lo largo del proyecto, la solución al problema analizando no pasa por utilizar tecnologías big data, este planteamiento nos indica el desconocimiento actual que el mundo empresarial tiene sobre estas tecnologías. A raíz de esto el proyecto intenta arrojar un poco de luz sobre el estado tecnológico actual en las empresas españolas y más en profundidad, de las empresas del sector de distribución alimentaria, gracias a la experiencia laboral personal.

1.2 OBJETIVOS

El principal objetivo del proyecto pretende mostrar un caso real de una empresa española del sector de distribución alimentaria, en la que surge una necesidad tecnológica relacionada con el problema de funcionamiento del actual sistema utilizado como almacén de datos.

Para alcanzar este objetivo principal se proponen los siguientes objetivos parciales:

- A raíz del desconocimiento de tecnologías Big Data, se pretende mostrar el estado del entorno tecnológico relacionado con la gestión de los datos, exponiendo la visón conceptual y teórica del término Big Data y la situación real de las empresas españolas del sector en base a la experiencia profesional personal

- Explicar el funcionamiento de los sistemas Data Warehouse así como mostrar tres de las opciones disponibles en el mercado pertenecientes a los fabricantes mejor posicionados.

- Explicar el funcionamiento de las herramientas de Extracción, Transformación y Carga de datos (ETL) así como mostrar tres de las opciones disponibles en el mercado pertenecientes a los fabricantes mejor posicionados.

- Mostrar la problemática real de una empresa del sector de distribución alimentaria, explicando el proceso de toma de decisiones de la solución escogida.

- Explicar las características y funcionamiento de los componentes hardware y software elegidos como solución al problema planteado.

- Describir un caso de uso de transformación de un trabajo de carga de datos a la nueva plataforma desplegada.

1.3 FASES DEL PROYECTO

Para poder realizar el proyecto ha sido necesario ir completando una serie de fases:

- Fase de Análisis:
    - Análisis del estado del arte relacionado con la gestión del dato
    - Búsqueda de las propuestas disponibles en el mercado de Data Warehouse
    - Búsqueda de las propuestas disponibles de herramientas ETL
    - Análisis del problema existente en una empresa del sector de distribución alimentaria
- Fase de Desarrollo
    - Explicación de la Prueba de Concepto desarrollada para la elección de la solución tecnológica.
    - Arquitectura sistema Data Warehouse
    - Arquitectura sistema ETL
    - Análisis y diseño de caso de uso presentado como ejemplo
- Fase de Documentación
    - Recopilación de datos
    - Generar la memoria del proyecto

1.4 HERRAMIENTAS EMPLEADAS

A continuación se realiza una breve descripción de las herramientas utilizadas en el desarrollo del proyecto, tanto en el aspecto técnico como en la redacción de la memoria del mismo.

Microsoft Office 2013

Microsoft Office 2013 cuenta con herramientas para editar textos, realizar hojas de cálculo, presentaciones de diapositivas y otras muchas aplicaciones. La elaboración de la presente memoria se ha realizado mediante Microsoft Word, la planificación del proyecto se ha realizado con Microsoft Project y los costes del proyecto se han calculado mediante Microsoft Excel.

InfoSphere DataStage Designer

[InfoSphere DataState Designer] proporciona las herramientas necesarias para crear trabajos que extraen, transforman, cargan y comprueban la calidad de los datos.

El cliente del Diseñador es como un entorno de trabajo o un lienzo blanco que se utiliza para crear trabajos. Esta herramienta tiene una paleta que contiene las herramientas que constituyen los pilares básicos de un trabajo:

- Las etapas se conectan a orígenes de datos para leer o grabar archivos y para procesar datos.
- Los enlaces conectan las etapas por las que los datos fluyen.
- Las anotaciones proporcionan información sobre los trabajos que se crean.

Además, utiliza un repositorio donde se pueden almacenar los objetos creados durante el proceso de diseño. Estos objetos pueden ser reutilizados por otros diseñadores de trabajos.

InfoSphere Data Architect

[IBM InfoSphere Data Architect] es una solución de diseño de datos de colaboración. Permite descubrir, modelar, relacionar, estandarizar e integrar activos de datos diversos y distribuidos en toda la empresa. IBM InfoSphere Data Architect facilita la comprensión de activos de datos actuales para incrementar la eficiencia y reducir el tiempo de comercialización.
InfoSphere Data Architect permite:

- Incrementar la eficiencia y reducir el tiempo de comercialización con una mayor comprensión de los activos de datos y trabajando de forma más eficiente con modelos de datos.
- Simplificar las tareas de diseño de Data Warehouse, modelado dimensional y gestión de cambios para lograr un desarrollo más rápido y sencillo.
- Mejorar la integración con productos relacionados para incrementar la colaboración.

Aginity Workbench for Netezza

[Aginity Workbench Netezza] ofrece una amplia gama de capacidades de gran alcance para hacer la gestión de un Data Warehouse de Netezza (IBM Pure Data) más eficiente con menos esfuerzo.

El producto comprende capacidades que permiten la construcción de una línea de comandos con una interfaz gráfica fácil de usar, el registro de los resultados de ejecución de SQL a una base de datos externa, y es capaz de completar las sentencias SQL de forma automática.

PuTTY

[PuTTY] es un cliente SSH, Telnet, rlogin, y TCP raw escrito y mantenido principalmente por Simon Tatham, open source y licenciado bajo la Licencia MIT.

1.5 ESQUEMA DE LA MEMORIA

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo:

- **Resumen**: en esta sección se incluye un resumen del proyecto, así como un listado de palabras clave por las que se puede identificar.
- **Abstract**: en esta sección se traduce la sección anterior al inglés.
- **Índices**: sección en la que se incluye un índice general del documento, así como un índice de figuras y de tablas.
- **Capítulo 1 - Introducción**: se introduce la memoria del proyecto fin de carrera detallándose sus objetivos y explicando las fases de desarrollo del proyecto brevemente. Además se detallan los medios utilizados y se realiza un breve resumen de cada uno de los apartados de la memoria.
- **Capítulo 2 - Estado del arte:** se realiza un resumen de lo más destacable durante la investigación sobre Big Data, como concepto y su implantación real en la empresa española, sistemas Data Warehouse, herramientas ETL, y se analiza y expone el problema específico que se pretende solucionar.
- **Capítulo 3 - Solución propuesta:** se realiza una exposición de la solución propuesta al problema planteado, explicando la arquitectura hardware y software del Data Warehouse y de la herramienta ETL así como el caso de uso detallado.
- **Capítulo 4 - Conclusiones**: en esta sección se detallan las conclusiones obtenidas tras la realización del proyecto, así como las mejoras futuras propuestas.
- **Capítulo 5 - Presupuesto**: en esta sección se detalla la planificación del proyecto incluyendo un diagrama de Gantt y presupuesto.
- **Capítulo 6 - Glosario**: en esta sección se incluyen todos los acrónimos utilizados en el documento y se indica el significado de sus siglas.
- **Capítulo 7 - Referencias:** en esta sección se detallan todas las referencias utilizadas durante la investigación y aquellas que son mencionadas a lo largo del presente documento.

2 ESTADO DEL ARTE

2.1 ¿QUÉ ES BIG DATA?

2.1.1 BIG DATA COMO CONCEPTO

Aunque el término "Big Data" se ha convertido en una palabra muy utilizada en cualquier foro, publicación, debate o "tweets" no tenemos que dejarnos engañar y creer que es una moda pasajera como tantas otras que acontecen en el mundo de las empresas de Tecnologías de la Información.

Según podemos extraer de la encuesta realizada por IDG Enterprise – International Data Group es una empresa especializada en publicaciones técnicas, eventos y páginas webs sobre tecnologías de la información-, el "boom" asociado al término "Big Data" es mucho más real de lo que parece.

De su reciente encuesta sobre "Big Data y Analítica 2015" [IDG 2015] podemos reseñar los siguientes aspectos que nos demuestras que el mundo empresarial está apostando por los nuevos paradigmas que nos presenta el mundo "Big Data"

- A lo largo del 2014, el número de organizaciones con proyectos de Big Data en desarrollo o ya desplegados se incrementó un 125%.
- Las grandes organizaciones invierten significativamente más capital en iniciativas centradas en el "dato" que lo que realizan las pymes; 13,8 millones de dólares frente a 1,6 millones de dólares.
- La mayoría de las organizaciones tiene previsto invertir en análisis de datos ya que esperan obtener el máximo valor de negocio de estas soluciones.
- La confianza en las soluciones de seguridad de los datos ha aumentado de un 49% en 2014 a un 66% en 2015
- Mientras que la confianza en las soluciones de seguridad aumenta, las organizaciones también se dan cuenta que es necesario proteger los datos de su empresa ya que cada vez más el dato se está convirtiendo en un activo con un valor significativo.
- La demandas de empleos relacionados con Arquitectos y Analistas de Datos se ha disparado durante los últimos 2 años

Después de analizar estos datos podemos llegar a la conclusión que la "revolución del dato" es real y está afectando o va a afectar a la mayoría de las empresas del mundo.

2.1.1.1 ¿Qué es "Big Data"? ¿De dónde viene ese término?

Par poder entender bien cuál es la revolución que estamos viviendo en el mundo tecnológico relacionada con el "dato" tenemos que tener una visión cronológica de la evolución que ha tenido este entorno en los últimos años

1970 Base de datos relacional

En 1970, Edgar F. Codd, matemático que realizó sus estudios en la universidad de Oxford y que se encontraba trabajando en el "IBM Research Lab", publicó un artículo en el que explicaba la forma en la que se podía accederse a la información almacenada en bases de datos de gran tamaño sin saber cómo estaba estructurada la información o donde residía dentro de la base de datos. Hasta este momento el método para extraer dicha información se necesitaba de unos conocimientos informáticos muy complejos que provocaba una fuerte inversión en tiempo y personal cualificado. Hoy en día todas las transacciones que se realizan diariamente utilizan estructuras basadas en la teoría de la base de datos relacional

1976 Sistemas de Planificación de necesidades de material (MRP)

A mediados de la década de 1970, los sistemas de Planificación de necesidades de material (MRP) se diseñaron como herramienta que ayudaba a las empresas de fabricación a organizar y planificar su información. Esta transformación marcó un cambio de tendencia hacia los procesos de negocio y las funcionalidades de contabilidad, y en este ámbito se fundaron empresas como Oracle, JD Edwards y SAP. Fue Oracle la que presentó y comercializó el Lenguaje de consulta estructurado o Structure Query Language (SQL) original.

1983 Crecimiento de la información impulsado por el sector de la comunicación

Durante la década de los 80s los avances tecnológicos permitieron que todos los sectores se beneficiaran de las nuevas formas de organizar, almacenar y genera datos. Estas mejoras provocaron que las empresas comenzaran a utilizar sus datos para mejorar la toma de decisiones en su negocio. El constante crecimiento en volumen de los datos generados y consumidos se debe al fuerte aporte del sector de las comunicaciones ya que durante esta década sufren su mayor expansión y penetración en la sociedad.

1985 La necesidad de obtener datos de calidad

En 1985, Barry Devlin y Paul Murphy, trabajadores de IBM, definieron una arquitectura para los informes y análisis de negocio que se convirtió en la base del almacenamiento de datos. En el centro de dicha arquitectura, se encuentra la necesidad del almacenamiento homogéneo y de alta calidad de datos históricamente completos y exactos.

1988 Nuevos sistemas software y hardware

Según avanza la década de los 80s se puede presenciar el auge de los sistemas de planificación MRP y el nacimiento de los sistemas de Planificación de recursos empresariales – también llamados ERP. Esta evolución, en conjunto con la mejora de los sistemas de almacenamiento de los datos provoca una demanda cada vez mayor de datos y un volumen en constante crecimiento.

1989 Inteligencia empresarial

En 1989, Howard Dresner amplió el popular término genérico "Business Intelligence (BI)" o Inteligencia empresarial, inicialmente acuñado por Hans Peter Luhn en el año 1958. Dresner lo definió como los "conceptos y métodos que mejoran la toma de decisiones de negocio mediante el uso de sistemas de apoyo basados en datos reales". Poco tiempo después, y como respuesta a la necesidad de una mejor BI, pudimos ver el auge de empresas como Business Objects, Actuate, Crystal Reports y MicroStrategy, que ofrecían informes y análisis de los datos de las empresas.

1992 El primer informe de base de datos

En 1992, Crystal Reports creó el primer informe de base de datos sencillo con Windows. Estos informes permitían a las empresas crear un informe sencillo a partir de diversos orígenes de datos con escasa programación de código. De esta forma, se redujo la presión existente sobre el panorama saturado de datos, y se permitió que las empresas emplearan la inteligencia empresarial de un modo asequible.

1995 Explosion de la Word Wide Web

En la década de los 90s se produjo un crecimiento tecnológico asombroso que provocó una generación masiva de nuevos datos en formatos no heterogéneos que necesitaban poder tratarse para obtener beneficio en el negocio. Los datos de inteligencia empresarial comenzaban a poder tratarse con sistemas más accesibles al usuario final como era el software de Microsoft Excel.

1997 "Big Data"

La primera vez que pudimos leer el término "Big Data" fue en un artículo de los investigadores de la NASA Michael Cox y David Ellsworth. En el afirmaban que el ritmo de crecimiento de los datos empezaba a ser un gran problema para los sistemas informáticos de la época.

1999 Se cuantifica la información

Debido al auge de las Word Wide Web y de los sistemas de inteligencia empresarial la cantidad de datos nueva y original que se crea en un año asciende a 1,5 exabytes – 1 EB $=1000^6$bytes$=10^{18}$bytes = 1000000000000000000B = 1000 petabytes = 1millionterabytes = 1billiongigabytes

2001 "Las Tres V"

Las diferencias básicas entre las aplicaciones desarrolladas hasta esta fecha y los nuevos conceptos de Big Data se basan en tres términos: Volumen, Variedad y Velocidad.

Volumen, se haba de Big Data cuando los volúmenes superan la capacidad del software habitual para ser manejados y gestionados, estamos hablando de Terabytes, Petabytes o Exabytes.

Variedad, este concepto aborda la inclusión de nuevas fuentes de datos, diferentes a las tradicionales, información obtenida de Redes Sociales, dispositivos electrónicos, sensores… en definitiva orígenes de diversas fuentes. Esta información a diferencia de los sistemas Data Warehouse tradicionales, puede estar semiestructurada o no tener estructura alguna.

Velocidad, este concepto es clave en los sistemas Big Data, la velocidad a la que se reciben los datos, se procesan y se toman decisiones es esencial para sistemas en tiempo real como la detección de fraude o la creación de ofertas personalizadas.

2005 La gestión de la base de datos, el centro del universo

Tim O'Reilly afirma en su informe What Is Web 2.0 que el SQL es el nuevo HTML. La gestión de las bases de datos es una tarea básica de las empresas Web 2.0, por lo que los datos se vuelven el centro neurálgico del desarrollo de webs y aplicaciones.

2006 Solución de código abierto para Big Data: HADOOP

Ante la necesidad de gestionar los sistemas de información que cada vez más explotaban datos obtenidos de la web, en 2006 se creó HADOOP. Software de código abierto, permite el procesamiento en paralelo y distribuido de un volumen de datos enorme en servidores de bajo coste y fácil escalabilidad.

2009 La Inteligencia Empresarial (BI) se convierte en la mayor prioridad

En el año 2009, la inteligencia empresarial (BI) y todas las herramientas asociadas a ella se convierte en la prioridad de mayor nivel para los directores de tecnologías de la información.

2011 Las grandes empresas amplían sus sistemas de almacenamiento de datos

El tamaño de los sistemas de almacenamiento de datos de las grandes empresas americanas crece cada año a mayor velocidad. Se calcula que las grandes empresas guardaron 7,4 Exabytes de datos originales.

2013 Avances tecnológicos en alza

Sistemas Data Warehouse cada vez más potentes, bases de datos en memoria, nuevos lenguajes de programación para entornos Big Data. Las empresas cada vez invierten más en recursos tantos físicos como intelectuales.

2020 El futuro del Big Data

La producción de datos aumenta a un ritmo espectacular. Los expertos apuntan a un aumento estimado del 4300 % en la generación de datos anuales para 2020. Entre los principales motivos que llevan a este cambio se incluyen el cambio de tecnologías analógicas a digitales y el rápido aumento en la generación de datos, tanto por particulares como por grandes empresas

2.1.2 BIG DATA EN EL ENTORNO PROFESIONAL ESPAÑOL

Durante el año 2012 España sufrió la propagación y expansión del término Big Data proveniente de un mercado americano donde la tecnología y los beneficios de su uso estaban mucho más maduros. Los medios de comunicación tanto especializados como generalistas ayudaron a que se generase un "hype" que es como se conocen aquellos conceptos que disfrutan de una cobertura por parte de la prensa que, a veces, no se corresponde con su valor real.

Este enorme interés en el concepto Big Data provocó que comenzaran a llegar a los círculos especializados las nuevas tecnologías que rodean al mundo Big Data, como era HADOOP y las bases de datos NoSQL que nos querían mostrar las herramientas necesarias para poder procesar una enorme cantidad de datos, de fuentes y contenidos diversos a una gran velocidad.

Ante tanta expectación, las grandes empresas comenzaron a formar grupos de trabajo que se encargarían de realizar pilotos de estas tecnologías con el fin de extraer el verdadero potencial de su uso.

Dado que no se disponía de casos de éxito conocidos o de experiencias de competidores en el sector, optaron por comenzar con las que a priori son las enormes generadoras de datos de nuestra época: Las Redes Sociales. Entorno a esta idea surgieron varios pilotos con la idea de explotar los datos de Twitter debido a las facilidades de acceso que proporciona.

Una vez superada la "emoción" inicial, se pudo comprobar que las ideas revolucionarias y futuristas que planteaba el Big Data no eran tan sencillas de poner en práctica. Casi toda la tecnología se basaba en software Open Source lo que provocó que la gente conocedora de su funcionamiento y con capacidades para aplicarlo era escasa. Era necesario invertir muchos recursos en formar un equipo capaz de implementar un piloto utilizando estas tecnologías.

Debido a estas complicaciones, los usuarios de negocio no percibían el valor que esta tecnología podía aportar a los casos planteados y la gestión del cambio necesario para adaptarse a los nuevos procedimientos no se había valorado correctamente.

Llegados a este punto las grandes empresas se plantearon que efectivamente era necesario mejorar los procesos de explotación de sus datos pero que primero era necesario modernizar, implementar o cambiar los sistemas de almacenamiento de datos (Data Warehouse) y las herramientas de transformación y carga de esos datos (ETL´s).

¿Por qué invertir primero en algo que ya conozco? En el caso de las grandes compañías que ya disponían de entornos completos de transformación de datos y almacenamiento de los mismos se estaban encontrando con que cada vez generaban más cantidad de datos, esto impactaba directamente en el retraso de los procesos de transformación de los mismos lo que provoca que la frescura del dato no fuera la idónea para alimentar correctamente a las herramientas de inteligencia empresarial (Bussines Inteligence).

A lo largo del 2013 y 2014 las grandes empresas consumidoras de estas tecnologías fueron implantando los últimos avances y en paralelo creando equipos de trabajo para abordar los proyectos Big Data a los que si veían futuro una vez solucionado los problemas de los "sistemas tradicionales"

En 2015 no podemos negar que en España se están realizando proyectos de Big Data, si bien es verdad que por ahora solo se lo pueden permitir grandes empresas a las que el conocimiento de los sentimientos que sus clientes tienen hacia ellas les proporciona una ventaja en su negocio. Estamos hablando de grandes bancos como BBVA, Bankia, Banco Santander, empresas de telecomunicaciones como Telefónica y Vodafone, alguna empresa de seguros están comenzando a interesarse en estas tecnologías para por ejemplo hacer un seguimiento del fraude de sus asegurados.

En la otra cara del mercado están todas las instituciones dedicadas a la investigación científica, como proyectos del CERN o la NASA.

En el ámbito de la distribución alimentaria, en el que centraremos el proyecto, desde hace 2 años se comenzó a invertir en tecnologías para modernizar los sistemas Data Warehouse tradicionales. Se decidió apostar primero en esta modernización ya que la toma de decisiones en este negocio sí que se iba a ver beneficiada al tener un análisis de datos mucho más potente que el disponible hasta la fecha. Hasta la fecha, la lentitud en los procesos de integración y análisis de los datos provocaba una tardía respuesta en ámbitos como aprovisionamiento de stock de alimentos, análisis de la cesta de la compra, fidelización de cliente, campañas de marketing…

Conceptos de Big Data como grandes volúmenes de datos no homogéneos y provenientes de diversas fuentes (Redes Sociales, foros, webs), todavía no han obtenido un valor suficiente para los usuarios de negocio como para ver la necesidad de realizar proyectos sobre estas áreas. Este sector no tiene una fuerte penetración en estas tecnologías como para poder obtener un conjunto de datos maduros de los que obtener resultados fiables como para tomar decisiones de negocio con ellos.

2.2 SISTEMAS DATA WAREHOUSE

De acuerdo con la definición de W.H. Inmon [Inmon W 1992] - creador del concepto – "un Data Warehouse es un conjunto de datos integrados orientados a una materia, que varían con el tiempo y que no son transitorios, los cuales soportan el proceso de toma de decisiones de una administración".

El punto diferenciador entre las bases de datos operacionales y el Data Warehouse es que este último reúne información de varias fuentes. Esta recopilación de datos de diversas fuentes, trasciende a través del tiempo y permite almacenar en un único lugar la información generada en distintos momentos del tiempo por distintas aplicaciones de software, que a su vez han utilizado distintas tecnologías de almacenamiento y variadas técnicas de gestión de bases de datos.

El nivel de detalle que presentan las bases de datos tradicionales suele no ser el adecuado para apoyar la toma de decisiones. El Data Warehouse, debe reunir esos datos y asociar otros, para presentar la información de forma tal que sirva como soporte de decisiones.

Parte de la información que utiliza el Data Warehouse proviene de datos históricos que son almacenados en favor de su eliminación de los sistemas operacionales ya que estos no son necesarios para las aplicaciones transaccionales. Debido a esta circunstancia el volumen de datos que almacena un sistema Data Warehouse es mayor que los sistemas operacionales.

La ventaja principal de este tipo de sistemas se basa en su concepto fundamental, la estructura de la información. Este concepto significa el almacenamiento de información homogénea y fiable, en una estructura basada en la consulta y el tratamiento jerarquizado de la misma, y en un entorno diferenciado de los sistemas operacionales.

Un sistema Data Warehouse se caracteriza por ser: [Carlos Fernández, Dataprix]

- ***Integrado***: los datos almacenados en el Data Warehouse deben integrarse en una estructura consistente, por lo que las inconsistencias existentes entre los diversos sistemas operacionales deben ser eliminadas. La información suele estructurarse también en distintos niveles de detalle para adecuarse a las distintas necesidades de los usuarios.

- ***Temático***: sólo los datos necesarios para el proceso de generación del conocimiento del negocio se integran desde el entorno operacional. Los datos se organizan por temas para facilitar su acceso y entendimiento por parte de los usuarios finales. Por ejemplo, todos los datos sobre clientes pueden ser consolidados en una única tabla del Data Warehouse. De esta forma, las peticiones de información sobre clientes serán más fáciles de responder dado que toda la información reside en el mismo lugar.

- ***Histórico***: el tiempo es parte implícita de la información contenida en un Data Warehouse. En los sistemas operacionales, los datos siempre reflejan el estado de la actividad del negocio en el momento presente. Por el contrario, la información almacenada en el Data Warehouse sirve, entre otras cosas, para realizar análisis de tendencias. Por lo tanto, el Data Warehouse se carga con los distintos valores que toma una variable en el tiempo para permitir comparaciones.

- ***No volátil***: el almacén de información de un Data Warehouse existe para ser leído, y no modificado. La información es por tanto permanente, significando la actualización del Data Warehouse la incorporación de los últimos valores que tomaron las distintas variables contenidas en él sin ningún tipo de acción sobre lo que ya existía.

Otra característica importante de un sistema Data Warehouse [Sinnexus] es que contiene metadatos, es decir, datos relativos a los datos. Los metadatos permiten saber la procedencia de la información, su periodicidad de refresco, su fiabilidad, forma de cálculo... etc.

Los metadatos serán los que permiten simplificar y automatizar la obtención de la información desde los sistemas operacionales a los sistemas informacionales.

Los objetivos que deben cumplir los metadatos, según el colectivo al que van dirigido, son:

- Dar soporte al usuario final, ayudándole a acceder al Data Warehouse con su propio lenguaje de negocio, indicando qué información hay y qué significado tiene. Ayudar a construir consultas, informes y análisis, mediante herramientas de Business Intelligence.

- Dar soporte a los responsables técnicos del Data Warehouse en aspectos de auditoría, gestión de la información histórica, administración del sistema, elaboración de programas de extracción de la información, especificación de las interfaces para la realimentación a los sistemas operacionales de los resultados obtenidos... etc.

Principales ventajas de utilizar sistemas Data Warehouse en el entorno empresarial:

- **Mejora de las herramientas de toma de decisiones**, Los beneficios se obtendrán mediante la mejora del acceso a la información. Gerentes y ejecutivos serán liberados de tomar decisiones basadas en datos limitados o en propias "corazonadas". Las decisiones que afectan a la estrategia y las operaciones de las organizaciones se basaran en hechos creíbles y serán respaldados con pruebas y datos reales. Por otra parte, los "tomadores de decisiones" estarán mejor informados, ya que podrán consultar datos reales y recuperar información en base a sus necesidades personales.

- **Aumentar el rendimiento del sistema y de las consultas,** Los Data Warehouse están diseñados y construidos con un enfoque en la velocidad de recuperación de datos y análisis de los mismos. Por otra parte, también están diseñados para el almacenamiento de grandes volúmenes de datos y poder consultarlos rápidamente. Estos sistemas analíticos se construyen de manera diferente de los sistemas operativos que se centran en la creación y modificación de datos. En contraste, el almacén de datos se construye para el análisis y recuperación de datos en lugar de mantenimiento eficiente de los registros individuales (es decir, transacciones). Además, el almacenamiento de datos requiere de una carga del sistema grande por lo que es beneficioso que sea sacado del entorno operativo ya que distribuye eficazmente la carga de trabajo a través de la infraestructura de tecnología de toda la organización.

- **El acceso oportuno a los datos**, El Data Warehouse permite que los usuarios de negocio y tomadores de decisiones tengan acceso a los datos de muchas fuentes diferentes. Además, los usuarios de negocio tendrán que esperar poco tiempo en el proceso de recuperación de datos. Rutinas de integración de datos programadas, conocidas como ETL, se aprovecha de este entorno de almacenamiento de datos. Estas rutinas consolidan los datos de varios sistemas de origen y transformar los datos en un formato útil. Posteriormente, los usuarios de negocio pueden acceder fácilmente a los datos desde una única interfaz. Además, los consumidores de datos podrán consultar los datos directamente con menos apoyo de los departamentos de tecnología de la información. El tiempo, que los departamentos de tecnología de la información tienen que dedicar para desarrollar informes y consultas disminuye en gran medida ya que los usuarios de negocio tienen la capacidad de generar informes y consultas por su cuenta. El uso de herramientas de consulta y análisis contra un repositorio de datos consistente y consolidado permite a los usuarios de negocio dedicar más tiempo a la realización de análisis de datos y menos a la recopilación de datos.

- **Aprender de los datos del pasado para predecir situaciones futuras,** Los almacenes de datos contienen generalmente muchos años de datos que no puede ser almacenados dentro de sistemas transaccionales. Normalmente, los sistemas transaccionales, satisfacen la mayoría de requisitos de información de funcionamiento para un periodo de tiempo determinado, pero sin la inclusión de los datos históricos. Por el contrario, los sistemas Data Warehouse almacenan grandes cantidades de datos históricos y pueden permitir la inteligencia empresarial avanzada incluyendo el análisis de periodo de tiempo, análisis de tendencias, y predicción de tendencias. La ventaja del almacén de datos es que permite la presentación de informes avanzada y análisis de múltiples períodos de tiempo.

2.2.1 Principales soluciones Data Warehouse en el mercado profesional

En el mercado actual de sistemas Data Warehouse podemos centrar la atención en tres grandes fabricante, IBM, ORACLE y TERADATA. Entre los tres nos proporcionan las soluciones más punteras a nivel tecnológico y las que obtiene mayor confianza del consumidor.

2.2.2 IBM PURE DATA SYSTEM FOR ANALYTICS

IBM PureData for Analytics es un sistema de alto rendimiento, escalable, de ejecución asimétrica masivamente paralela (AMPP), que hace que los clientes de IBM cuenten con una plataforma analítica capaz de gestionar volúmenes de datos enormes. Es un dispositivo ofrecido en formato appliance, es decir, con todos sus componentes ya integrados, instalados y configurados de fábrica, que integra en su arquitectura una base de datos de altas prestaciones, el hardware servidor preciso para ejecutar el software embebido y el almacenamiento necesario.

Este sistema, basado en tecnología de Netezza, ha sido diseñado específicamente para ejecutar cargas analíticas muy complejas sobre volúmenes ingentes de datos, pero de una manera muy sencilla, haciendo que los costes de mantenimiento y operación sean mucho menores que con otros sistemas existentes.

Ilustración 2 - IBM Pure Data System For Analytics

La base del diseño de IBM Pure Data System for Analytics es la de tratar de eliminar en la medida de lo posible, el movimiento de datos a lo largo del sistema, llevando la ejecución de los procesos analíticos a donde residen los datos, en lugar de realizar el camino contrario.

Gracias a esto, el rendimiento ofrecido por PureData for Analytics permite realizar análisis sobre los datos que hasta ahora no eran posibles de realizar, al tiempo que se mejoran los rendimientos de las tareas existentes.

IBM Pure Data System for Analytics es una solución construida específicamente para resolver este tipo de cargas analíticas, basado en estándares de gestión de sistemas de Data Warehouse, integrando en un solo producto el software de base de datos, los servidores hardware, el almacenamiento y las capacidades analíticas avanzadas, es decir, se proporciona una solución integrada y completa, preinstalada y preconfigurada, que no requiere tuning adicional para conseguir el rendimiento ofrecido por el sistema.

Además de un software específicamente diseñado para este tipo de cargas de trabajo, una buena parte de la capacidad de ofrecer un rendimiento superior en varios órdenes de magnitud reside en un revolucionario diseño hardware. De este modo, su potencia no proviene del uso de los componentes más caros existentes en el mercado, sino de cómo los componentes estándar utilizados en su desarrollo se integran correctamente para maximizar su potencial y ofrecer el rendimiento sin igual que presenta la plataforma.

De este modo, el Procesamiento Asimétrico Masivo en Paralelo (AMPP), combina múltiples CPUs Intel estándar con los procesadores FPGA, componentes también estándar de mercado (presentes por ejemplo en los reproductores de DVD), pero utilizados de manera única por el sistema, lo que hace que la lectura de datos del disco se minimice y al tiempo acelere, consiguiendo el mencionado rendimiento superior.

Por otro lado, la increíble facilidad de uso del appliance propuesto, PureData for Analytics, proporciona unos resultados óptimos sin necesidad de crear índices o realizar tuning constantemente sobre la plataforma.

La puesta en marcha se realiza en horas, no semanas, y no hay necesidad de realizar una administración costosa de la base de datos, permitiendo configuraciones completamente flexibles y sobre todo una escalabilidad lineal extrema, ya que se puede comenzar en el rango de unos pocos terabytes y hacer crecer la solución hasta varios petabytes, con los mismos costes de explotación y administración.

2.2.3 ORACLE EXADATA x5

Oracle Exadata x5 está diseñado para ser el sistema con mayor rendimiento, la plataforma más eficaz, la más segura y la más económica para el funcionamiento de la base de datos Oracle.

El sistema está formado por un conjunto de software y hardware diseñados para constituir la plataforma de máximo rendimiento y con la más alta disponibilidad para ejecutar Oracle Database. Su arquitectura cuenta con un diseño escalable con servidores estándar de la industria y almacenamiento inteligente, incluidas la tecnología flash de última generación y una estructura interna InfiniBand de alta velocidad. Las configuraciones elásticas permiten que los sistemas se adapten a cargas de trabajo de bases de datos específicas.

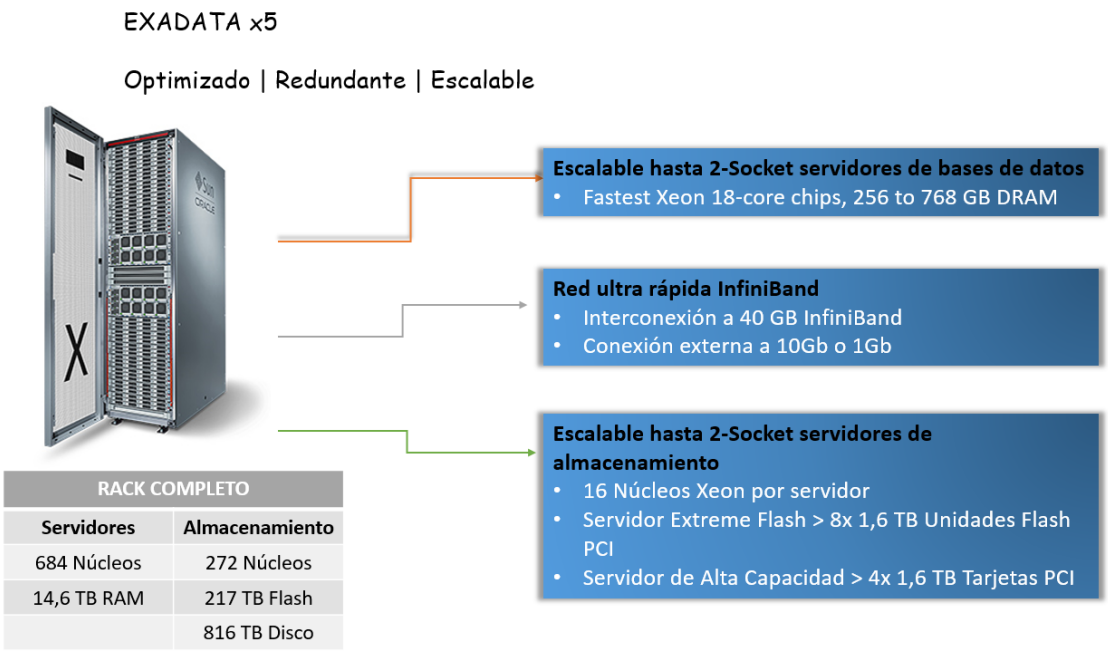

Ilustración 3 - Oracle Exadata x5

Exadata funciona con todos los tipos de cargas de trabajo de bases de datos que incluyen procesamiento de transacciones en línea (OLTP), almacenamiento de datos (DW), análisis en memoria y consolidación de cargas de trabajo mixtas.

Exadata es un sistema de fácil despliegue que incluye todo el hardware necesario para el funcionamiento de la base de datos Oracle. Los servidores de bases de datos, servidores de almacenamiento y de red están pre configurados, pre-tuneados y pre-probados por expertos de Oracle, de esta forma reducimos semanas o meses de esfuerzo para implementar un sistema de alto rendimiento.

Numerosas pruebas de extremo a extremo asegura que todos los componentes trabajan juntos sin problemas y no hay cuellos de botella de rendimiento o puntos únicos de fallo que pueden afectar el sistema completo.

Debido a que todos los sistemas Exadata están configurados de forma idéntica, los clientes se benefician de la experiencia de miles de otros usuarios que han implementado el sistema para sus aplicaciones. Los sistemas de los clientes también son idénticos a las máquinas de soporte que Oracle utiliza para la identificación y resolución de problemas, así como las máquinas que Oracle utiliza para el desarrollo y la prueba de la base de datos Oracle. Por lo tanto, Exadata es una plataforma probada a fondo y que dispone de un tunnig de la base de datos Oracle optimizado.

Principales características que proporciona el sistema Exadata:

CARACTERISTICA	DESCRIPCIÓN
Configuraciones elásticas	Ofrece un sistema específicamente optimizado para cargas de trabajo de base de datos que permite ahorrar en el presupuesto y, a la vez, optimizar el rendimiento para las necesidades de su base de datos
Opción de almacenamiento: Flash extremo	Ofrece el rendimiento todo flash optimizado de Oracle Database para todas las transacciones y consultas
Admite Oracle VM para la virtualización	Proporciona aislamiento para cargas de trabajo consolidadas que necesitan límites duros en CPU/ memoria o administración/

	sistema operativo independiente, y que pueden ser usadas por una máquina virtual para administrar opciones de base de datos de licencias y otro software
Diseñado para la base de datos de Oracle estándar	Las aplicaciones que actualmente utilizan Oracle Database pueden migrar sin problemas a Oracle Exadata Database Machine, sin necesidad de realizar cambios
Tecnología de Oracle Exadata Smart Scan	Mejora el rendimiento de las consultas trasladando el procesamiento de consultas intensivas y los puntajes de extracción de datos hacia servidores de almacenamiento inteligentes y escalables
Oracle Exadata Smart Flash Cache	Mejore los tiempos de respuesta de consultas y el rendimiento, almacenando de modo transparente en caché los datos más consultados para un almacenamiento rápido en estado sólido
Oracle Exadata Hybrid Columnar Compression	Mejora el rendimiento y disminuye los costos de almacenamiento, reduciendo el tamaño de las tablas de almacenamiento de datos hasta 10 veces, y el de las tablas de archivo, hasta 50 veces
Red InfiniBand	Le permite conectar múltiples Oracle Exadata Database Machine para formar una única configuración de imagen de sistema. Cada enlace de InfiniBand proporciona 40 GB de ancho de banda, muchas veces superior al almacenamiento tradicional o las redes de servidores tradicionales

Tabla 1 - Características Exadata x5

2.2.4 TERADATA DATA WAREHOUSE

El Sistema Teradata Data Warehouse es un appliance fabricado por la empresa Teradata que al igual que las soluciones descritas anteriormente, engloba en un único sistema nodos de gestión, nodos de almacenamiento y nodos de red.

Teradata Data Warehouse es una plataforma analítica flexible y rentable que puede ser utilizada como almacén de datos, data mart, sistema de recuperación o para pruebas y desarrollo. El dispositivo es una solución completa, integrada y preconfigurada que incluye el hardware Teradata Database, servidores administrados y tecnología de back-up opcional, todo en un solo sistema.

El Dispositivo ofrece un procesamiento más rápido in-memory y aumenta el rendimiento de consulta utilizando la tecnología Intel Haswell y los últimos componentes de memoria DDR4 para un acceso más rápido a los datos almacenados en la memoria.

Con una gran capacidad de potencia de procesamiento por terabyte de datos, sistema proporciona un rendimiento de consulta rápida y una gran escalabilidad. La arquitectura software "no compartir" ofrece siempre la ejecución en paralelo de las consultas, gracias a esto hasta las consultas más complejas puede completarse rápidamente.

Ilustración 4 - Teradata Data Warehouse Appliance

Las principales características que el sistema proporciona son:

Compresión automática de los datos

- Algoritmos de compresión de datos mejorados
- Posibilidad de aumentar el espacio disponible de la base de datos sin necesidad de realizar ninguna tarea administrativa

Optimizador de consultas paralelas

- Diseña el plan de consultas más rápido en consultas complejas sin necesidad de consejos
- Puede re-escribir consultas problemáticas en tiempo real para optimizar la respuestas frente a herramientas SQL o BI aprovechando toda la potencia del sistema

Configuración de alto rendimiento

- La posibilidad de tener índices primarios, índices multi-nivel particionados y índices de uniones de agregados permiten obtener resultados con mayor rapidez y sin la necesidad de escanear la tabla por completo
- Alto rendimiento debido al a interconexión de los dispositivos internos mediante redes InfiniBand

Completa gestión de carga de trabajo

- Capacidades de Gestión Integrada de la carga de trabajo mucho mayor gracias al sistema operativo Suse Linux Entrerprise Server v11
- Gestión de prioridad dinámico, incluyendo la posibilidad de priorizar la carga de trabajo en función de Acuerdo de nivel de servicio
- Si el sistema no está a pleno funcionamiento, la CPU se pone a disposición de los grupos de rendimiento de menor prioridad
- Los filtros y aceleradores del sistema ayudan a gestionar las consultas, las sesiones y las utilidades.

Funcionalidad Avanzada

- El sistema de control inteligente de memoria de Teradata, es capaz de en tiempo real monitorizar el uso de acceso a los datos para colocar los datos más accedidos en memoria. La memoria total disponible escala de forma automática y lineal a medida que el sistema crece debido a la tecnología de “no-compartir” nada que el sistema dispone.
- Disposición de datos en tablas columnares lo que provoca una mejora drástica en el rendimiento de las consultas y maximiza los ratios de compresión de los datos.
- Cifrado completo del dato opcional, a prueba de manipulaciones.

2.3 HERRAMIENTAS EXTRACCIÓN, TRANSFORMACIÓN Y CARGA (ETL)

ETL son las siglas en inglés de Extraer, Transformar y Cargar (Extract, Transform and Load). Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos, limpiarlos, y cargarlos en otra base de datos, o Data Warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

Gracias a los procesos ETL es posible que cualquier organización:

- Mueva datos desde una o múltiples fuentes a uno o varios destinos.
- Reformatee esos datos y los limpie, cuando sea necesario.
- Los cargue en un destino como puede ser una base de datos, un data mart o un Data Warehouse.
- Una vez alojados en destino, los datos serán analizados por herramientas de inteligencio de negocio
- Posea un control de la extracción de los datos y su automatización, disminuyendo el tiempo empleado en el descubrimiento de procesos no documentados y permitiendo mayor flexibilidad en el desarrollo.
- Las herramientas proporcionan acceso a diferentes entornos que utilizan tecnologías heterogéneas lo que nos permite optimizar los recursos empleados en el desarrollo
- Uso de la arquitectura de metadatos, facilitando la definición de los objetos de negocio y las reglas de consolidación
- Planificación de trabajos, gestión de logs, interfaces para la integración con planificadores de terceros, que permitirán llevar una gestión de la planificación de todos los procesos necesarios para la carga de los datos.
- Interfaz independiente de hardware

Los procesos ETL también se pueden utilizar para la integración con sistemas heredados (aplicaciones antiguas existentes en las organizaciones que se han de integrar con los nuevos aplicativos, por ejemplo, ERP´s. La tecnología utilizada en dichas aplicaciones puede hacer difícil la integración con los nuevos programas.

Ejemplo de flujo de datos a través de una herramienta ETL:

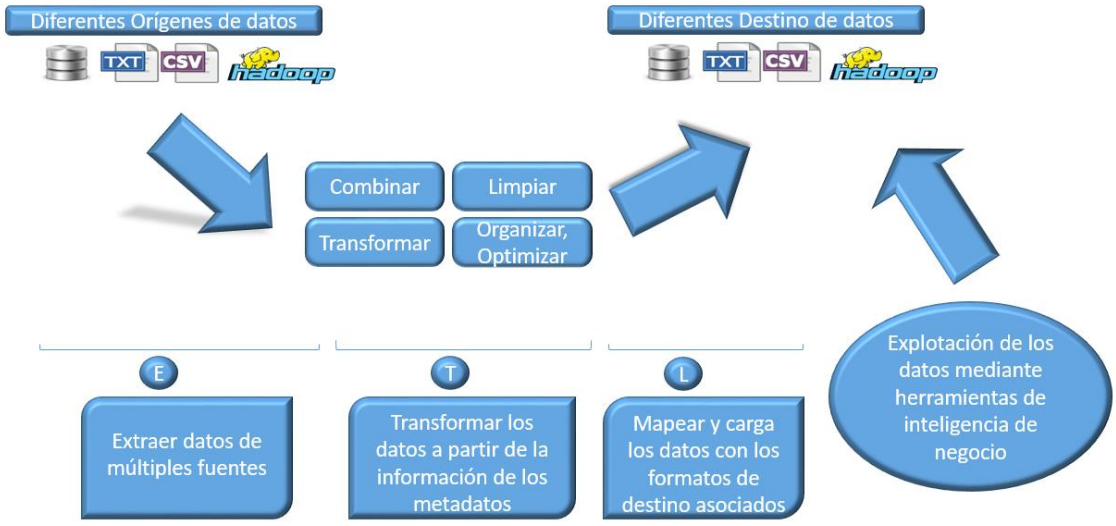

Ilustración 5 - Flujo de datos (ETL)

Como se ha comentado previamente, los procesos ETL constan de tres fases claramente diferenciadas, Extracción, Transformación y Carga.

Extracción

En esta fase se lleva a cabo el proceso de extracción de los datos desde los sistemas de origen. En la mayoría de los entornos los procesos integran datos provenientes de diferentes sistemas, con tecnologías heterogéneas. Cada sistema por separado, utiliza una organización de datos diferente. Los formatos de las fuentes que se suelen encontrar son, las bases de datos relacionales, ficheros de texto planos pero podemos encontrar bases de datos no relacionales u otras estructuras diferentes como sistemas ERP, Web Services, XML ... La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.

Para lleva a cabo la tarea de extracción de datos tenemos que seguir unos pasos básicos:

- Extraer los datos desde los sistemas de origen.
- Analizar los datos extraídos obteniendo un chequeo de la calidad de los datos.
- Interpretar este chequeo para verificar que los datos extraídos cumplen los requisitos de calidad establecidos. Si no fuese así, los datos deberían ser rechazados.
- Convertir los datos a un formato preparado para iniciar el proceso de transformación.

Una exigencia importante que se debe tener en cuenta a la hora de ejecutar procesos de extracción de datos, es que estos causen el menor impacto posible en el sistema origen.
Si el volumen de los datos a extraer es elevado, el sistema origen podría verse afecto por el proceso de extracción, provocando una degradación en su rendimiento llegando incluso a provocar una pérdida del servicio. Por esta razón, en sistemas que manejan una procesos de extracción de grandes volúmenes de datos es necesario llevar a cabo una planificación de los trabajos en horas valle de utilización del sistema donde este impacto sea nulo o mínimo.

Transformación

A lo largo de la fase de transformación tiene lugar la implantación de los modelos de negocio desarrollados por la empresa, estos son utilizados sobre los datos obtenidos en la fase de extracción y tienen como resultado final la obtención de los datos en el formato de las fuentes de destino.
Adicionalmente a los modelos de negocio comúnmente es necesario aplicar funciones de transformación de datos más específicas, como por ejemplo:

- Limpieza de datos:
  - Eliminar datos no validos (nulos), corregir y completar datos, eliminar duplicados
  - Estandarización: codificación, formatos, unidades de medida
- Traducción de campos de códigos por sus descripciones
- Codificación de valores
- Transformación de formatos de campos
- Unión de datos por múltiples combinaciones
- Cálculos matemáticos
- Control de errores y generación de logs
- Planificación de ejecución de secuencias
- Envíos de notificaciones por correo electrónico

Carga

En la ejecución de esta fase, los datos previamente extraídos y transformados son cargados en los sistemas destino. En función de las necesidades de cada organización este proceso abarca acciones diferentes, en algunas bases de datos destino es necesario sobrescribir la información existente antes de realizar la carga de los nuevos datos.

En los sistemas Data Warehouse lo habitual es guardar un histórico de datos por lo que se tienen que añadir los nuevos a los ya existentes por lo que hay que tener en cuenta aspectos como, valores únicos, integridad referencial, rango de valores… para poder mantener los datos en el sistema en un estado coherente.

Tradicionalmente se describen dos procedimientos a la hora de realizar procesos de carga de datos: [Roberto Espinosa, DataPrix 2010]

- **Acumulación simple:** La acumulación simple es la más sencilla y común, y consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el Data Warehouse, almacenando un valor calculado que consistirá típicamente en un sumatorio o un promedio de la magnitud considerada.

- **Rolling:** El proceso de Rolling por su parte, se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada (por ejemplo, totales diarios, totales semanales, totales mensuales, etc.).

Desafíos para los procesos y Herramientas de ETL [Carlos Fernández, Dataprix]

Los procesos ETL pueden ser muy complejos. Un sistema ETL mal diseñado puede provocar importantes problemas operativos.

En un sistema operacional el rango de valores de los datos o la calidad de éstos pueden no coincidir con las expectativas de los diseñadores a la hora de especificarse las reglas de validación o transformación.

Normalmente los Data Warehouse son alimentados de manera asíncrona desde distintas fuentes, que sirven a propósitos muy diferentes. El proceso ETL es clave para lograr que los datos extraídos asíncronamente de orígenes heterogéneos se integren finalmente en un entorno homogéneo.

La escalabilidad de un sistema de ETL durante su vida útil tiene que ser establecida durante el análisis. Esto incluye la comprensión de los volúmenes de datos que tendrán que ser procesados según los acuerdos de nivel de servicio (SLA: Service Level Agreement).

El tiempo disponible para realizar la extracción de los sistemas de origen podría cambiar, lo que implicaría que la misma cantidad de datos tendría que ser procesada en menos tiempo. Algunos sistemas ETL son escalados para procesar varios terabytes de datos para actualizar un Data Warehouse que puede contener decenas de terabytes de datos.

El aumento de los volúmenes de datos que pueden requerir estos sistemas pueden hacer que los lotes que se procesaban a diario pasen a procesarse en micro-lotes (varios al día) o incluso a la integración con colas de mensajes o a la captura de datos modificados (CDC: change data capture) en tiempo real para una transformación y actualización continua.

2.3.1 Principales herramientas ETL disponibles en el mercado profesional

Dentro del amplio mercado disponible de herramientas ETL podemos fijar el objetivo en tres grandes fabricantes con una amplia experiencia en el sector: INFORMATICA, IBM, ORACLE.

IBM y ORACLE respaldan las herramientas con su amplia experiencia en Data Warehouse, Bases de datos tradicionales, sistemas Big Data… un portfolio de software relacionado con el dato inmenso. INFORMATICA por su parte es la pionera en las herramientas ETL y tiene años de experiencia en importantes implantaciones a nivel mundial.

2.3.2 INFORMATICA POWERCENTER

Informatica PowerCenter proporciona una plataforma global de integración de datos que permite acceder, descubrir e integrar datos desde múltiples sistemas heterogéneos, con formatos híbridos, para que posteriormente puedan ser distribuidos de forma sencilla y rápida.

Informatica PowerCenter proporciona una alta disponibilidad, un elevado rendimiento, una óptima escalabilidad, es decir, PowerCenter proporciona una de base de datos para desarrollar los proyectos de integración de una organización empresarial.

El principal objetivo de PowerCenter es integrar los datos de toda la empresa, facilitar la posibilidad de utilizar procesos con arquitectura SOA, proporcionar una gestión completa de flujo de los datos, así como de su integridad y calidad.

PowerCenter proporciona la información apropiada, en el momento exacto, para satisfacer los requisitos, tanto analíticos cómo operacionales.

PowerCenter ofrece:

- Acceso a los datos en tiempo real o por lotes, o de igual forma empleando la captura de cambios cómo CDC (Change Data Capture).
- PowerCenter Real Time Edition que gestiona procesos de integración de datos.
- Metadata Manager.
- PowerCenter Standard Edition (Table Manager, Mapping Architect for Visio y Mapping Architect for Excel).
- Herramientas de pruebas.

- Asistentes de configuración.
- Repository Manager: Tareas de administración del repositorio
- Designer: Diseño de las transformaciones
- Workflow Manager: Configuración del servidor y las ejecuciones
- Workflow Monitor: Revisión de ejecuciones
- Repository Server Administration Console: Administración del repositorio y de los servidores de repositorio

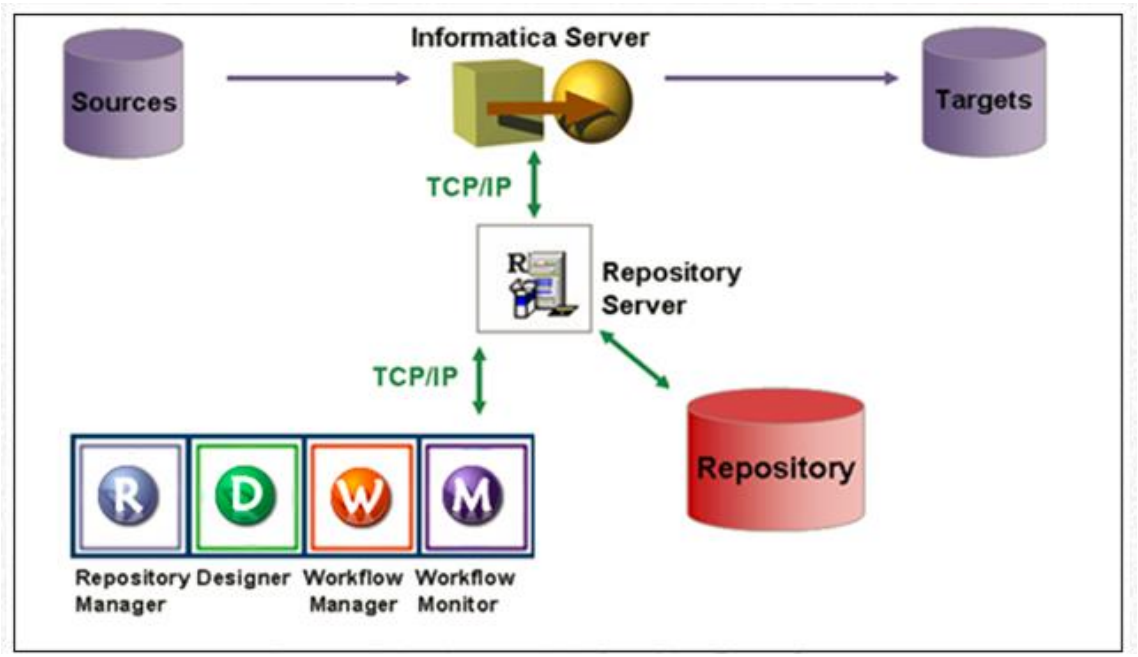

Ilustración 6 - Arquitectura Informatica PowerCenter [Juan Garcés 2013]

2.3.3 IBM INFOSPHERE DATASTAGE

InfoSphere DataStage es una herramienta de integración de datos que permite a los usuarios mover y transformar datos entre sistemas de fuentes diversas, de tipos de transacciones desiguales e incluso a sistema analíticos.

La transformación y el movimiento de datos es el proceso mediante el cual se seleccionan, convierten y correlacionan datos de origen en el formato que requieren los sistemas de destino. El proceso manipula datos para que sean conformes con las reglas de negocio, de dominio y de integridad y con otros datos en el entorno de destino.

InfoSphere DataStage proporciona conectividad directa a aplicaciones empresariales como orígenes o destinos, garantizando que los datos más relevantes, completos y precisos se integren en el proyecto de integración de datos.

InfoSphere DataStage integra los datos a través de múltiples sistemas que utilizan un marco paralelo de alto rendimiento, y es compatible con la gestión de metadatos extendido y conectividad empresarial. La plataforma escalable proporciona una integración más flexible de todos los tipos de datos, incluyendo grandes datos en reposo (con sede en Hadoop) o en movimiento (basado-stream), en las plataformas distribuidas y de mainframe.

Al utilizar las funciones de proceso paralelo de plataformas de hardware multiprocesador, InfoSphere DataStage permite a la organización resolver problemas empresariales a gran escala. Se pueden procesar grandes volúmenes de datos en un proceso por lotes, en tiempo real, o como un servicio web, en función de las necesidades del proyecto.

InfoSphere DataStage ofrece estas características y beneficios:

- Potente plataforma ETL escalable -Soporta la recolección, integración y transformación de grandes volúmenes de datos, con estructuras de datos que van de lo simple a lo complejo.
- El apoyo a los grandes datos y Hadoop que -permite acceder directamente a los grandes datos en un sistema de archivos distribuido, y ayuda a los clientes a aprovechar de manera más eficiente las nuevas fuentes de datos mediante el apoyo JSON y un nuevo conector JDBC.
- Acerca la integración de datos en tiempo real -así como la conectividad entre las fuentes de datos y aplicaciones.

- Gestión de carga de trabajo y las reglas de negocio: ayuda a optimizar la utilización del hardware y priorizar las tareas de misión crítica.
- Facilidad de uso: ayuda a mejorar la velocidad, flexibilidad y eficacia para construir, desplegar, actualizar y administrar la infraestructura de integración de datos.

Las aplicaciones cliente DataStage son comunes en todas las versiones:

- Administrador - Administra proyectos DataStage, gestiona la configuración global e interactúa con el sistema
- Diseñador - se utiliza para crear trabajos DataStage y secuencias de trabajo que se compilan en programas ejecutables. Se trata de un módulo principal para los desarrolladores.
- Director – gestiona la ejecución y seguimiento de los trabajos de DataStage. Se utiliza principalmente por los operadores y los probadores.
- Gerente - para gestionar, navegar y editar el repositorio de metadatos de almacenamiento de datos.

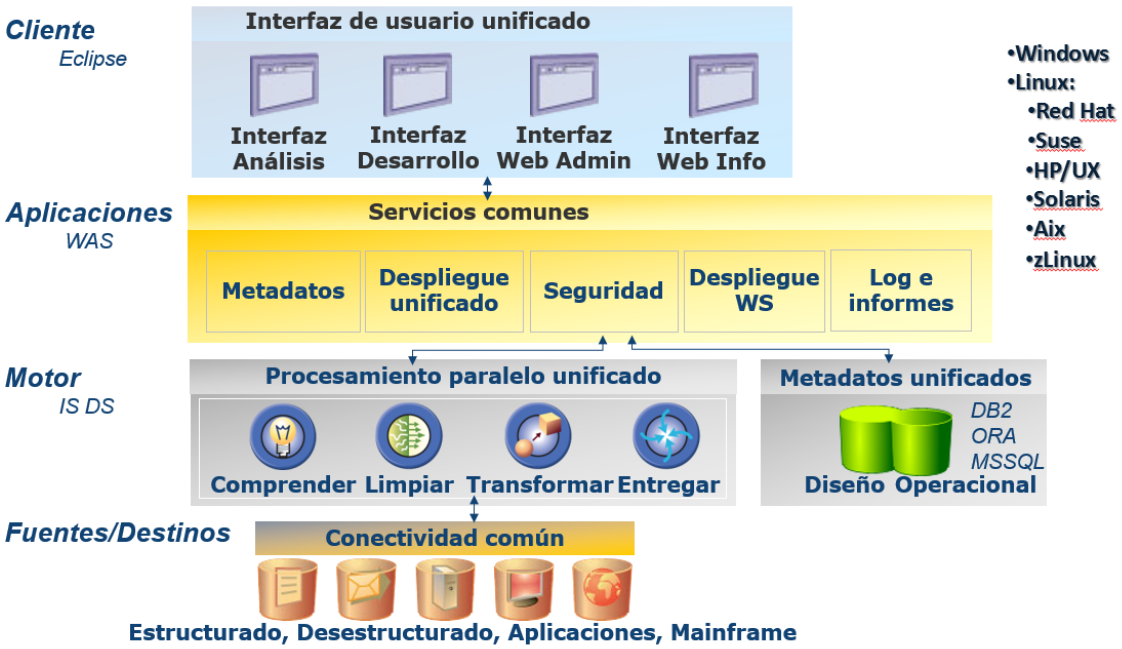

Ilustración 7 - Arquitectura IBM InfoSphere DataStage

2.3.4 ORACLE WAREHOUSE BUILDER

Oracle Warehouse Builder es una herramienta global que abarca todos los aspectos de la integración de datos. Warehouse Builder aprovecha la base de datos Oracle para transformar los datos en información de alta calidad. Proporciona la calidad de los datos, la auditoría de datos, modelado relacional y dimensiones totalmente integradas, y la gestión del ciclo de vida completo de datos y metadatos.

Warehouse Builder le permite crear almacenes de datos, migrar los datos desde los sistemas heredados, consolidar datos de fuentes de datos dispares, limpiar y transformar datos para proporcionar información de calidad, y administrar metadatos corporativos.

Muchas empresas disponen de datos dispersos en diferentes plataformas y utilizan una amplia variedad de herramientas de informes y análisis de datos. Los datos de clientes y proveedores pueden ser almacenados en aplicaciones, bases de datos, hojas de cálculo, archivos planos y sistemas heredados.

Esta diversidad puede ser causada por las unidades organizativas que trabajan de forma independiente durante un período de tiempo, o puede ser el resultado de las fusiones de negocios. Cualquiera que sea la causa de la diversidad, esta diversidad típicamente resulta en datos de mala calidad que proporcionan una visión incompleta e inconsistente de la empresa.

La transformación de datos de mala calidad en la información de alta calidad requiere:

- El acceso a una amplia variedad de fuentes de datos, Warehouse Builder aprovecha la base de datos Oracle para establecer conexiones transparentes a numerosas bases de datos de terceros, aplicaciones, archivos y almacenes de datos
- Capacidad de limpiar, transformar y depurar los datos, Warehouse Builder ofrece una amplia biblioteca de transformaciones de datos para tipos de datos tales como texto, numérico, fecha, y otros
- Capacidad para aplicar diseños para diversas aplicaciones, usando Warehouse Builder se puede diseñar e implementar cualquier almacén de datos requerido por las aplicaciones, ya sea relacional o con dimensiones
- Registros de auditoría, después de la consolidación de los datos de una variedad de fuentes en un único almacén de datos, es probable que sea necesario enfrentarse al desafío de la verificación de la validez de la información de salida. Warehouse Builder proporciona herramientas para poder abordar estas tareas

Funcionalidades diferenciales de Oracle Warehouse Builder:

Opciones de carga de datos avanzadas, Muchas empresas se enfrentan a la realidad de volúmenes de datos que crecen a un ritmo inmanejable, además el tiempo proceso para la carga de datos crecerá más y el tiempo de la ventana de carga será cada vez más corto. En muchos casos, es difícil completar las cargas de datos en las ventanas de tiempo disponibles utilizando técnicas de carga basadas en SQL.

Oracle Warehouse Builder automatiza la carga de datos usando métodos rápidos y eficientes, tales como Oracle Data Pump y tablespaces transportables. Mover datos a granel con estas técnicas resulta en gran medida en un aumento de rendimiento de carga debido a que la mayor parte del procesamiento SQL no se realiza.

Pluggable Mappings para reutilizar la lógica de transformación

Pluggable Mappings, permite a los desarrolladores reutilizar la lógica de las transformaciones que generen, son una extensión que se puede utilizar como operadores de otros mappings- similar a llamar a una subrutina o función en la mayoría de los lenguajes de programación. El resultado es que los desarrolladores pueden crear transformaciones complejas a partir de componentes más pequeños, reutilizarlos a través de un proyecto, y compartirlos entre los desarrolladores. Esto aumenta la productividad del desarrollador, reduce el trabajo y garantiza la coherencia de tareas complejas.

Gestión de la configuración avanzada

El software permite trasladar sistemas de un entorno a otro (por ejemplo, de un entorno de desarrollo a un entorno productivo). Cada ambiente tiene su propia información de conexión y configuración únicas para el paralelismo.

La función multi-configuración de Oracle Warehouse Builder permite definir un único diseño ETL lógico y asociarlo con muchos ambientes físicos separados.

A partir de un diseño común ETL, Oracle Warehouse Builder genera código ETL específicos para cada entorno físico en base a la configuración y la conexión de la información que se defina.

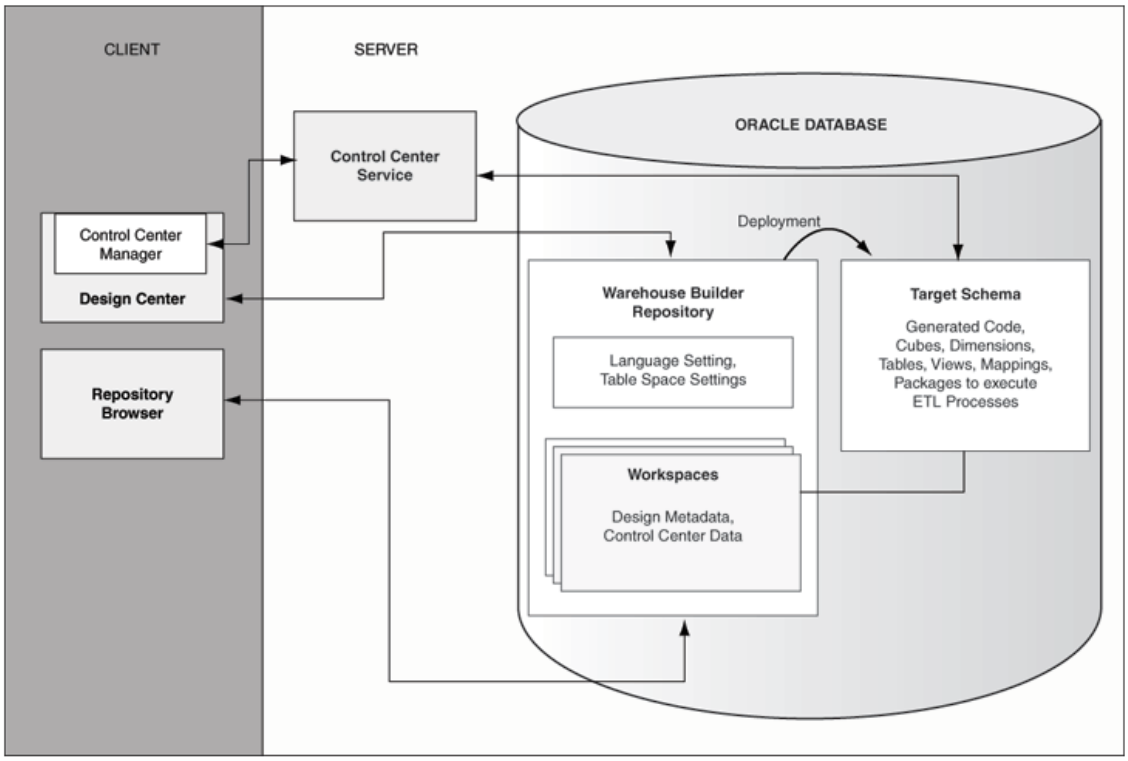

Ilustración 8 - Oracle Warehouse Builder components [Oracle 2015]

2.4 ANALISIS Y PROPUESTA DE SOLUCION DE PROBLEMÁTICA EN UN CLIENTE DEL SECTOR DISTRIBUCIÓN ALIMENTARIA

Vamos a proceder a explicar la problemática encontrada en un cliente del sector distribución alimentaria, relacionada con el control, gestión y optimización de sus datos que lleva a cabo con un sistema de almacén de datos ya implementado.

El cliente ha estado analizando el entorno de nuevas propuestas sobre tecnologías "Big Data" existentes en el mercado actual y se planteó abordar un proyecto de implantación de alguna de ellas para mejorar la gestión de la información de sus sistemas.

Realmente con un proyecto de Big Data no mejoramos la gestión de los sistemas ya existentes, pero el desconocimiento de este tipo de tecnología por parte de los clientes puede llevar a estas confusiones.

Una vez asesorado, el departamento de tecnologías de la información, se dio cuenta que antes de poder abordar un proyecto de "Big data" tenía que resolver el verdadero problema que llevaba arrastrando desde hace tiempo. El Data Warehouse ya existente, que era la única fuente de información para los usuarios de negocio.

"Antes de construir un tejado nuevo y más pesado, tengo que reforzar los cimientos de la casa"

El sistema actual, que tiene casi 10 años de antigüedad y está desarrollado mediante programas RPG y COBOL. Comenzaba a mostrar signos de un comportamiento no óptimo para los requisitos que se le exigían. Los datos residen en un sistema IBM iSeries con un comportamiento optimizado para entornos transaccionales pero no diseñado para albergar las cargas de trabajo de un Data Warehouse.

Alguno de los problemas existentes:

- Entorno de desarrollo estable pero costoso en recursos y tiempo de implantación de nuevas funcionalidades. Mantenimiento necesario por personal muy especializado.
- El funcionamiento de todo el proceso de "ETL" es puramente secuencial, restringido por el funcionamiento de los lenguajes de programación.
- Restricciones de datos en tablas provocados por la plataforma existente, límite de 4.000.000.0000 de registros por tabla en un sistema DB2 en iSeries, necesario particionar la tabla con el consiguiente aumento de la gestión.
- Mantenimiento de índices, vistas y estadísticas necesario.
- Demora en los procesos de carga de datos en el Data Warehouse, los tiempos de integración de los datos estaban en el rango de las 24 horas.

- Los tiempos de respuesta de las consultas realizadas por la herramienta de inteligencia de negocio no eran los óptimos, está circunstancia provocaba continuas incidencias con el usuario de negocio y la desconfianza en el sistema.
- Algunos procesos que se ejecutan periódicamente en fechas específicas y que tiene una gran carga de trabajo se demoran tiempos nada prácticos, hasta 12 días de ejecución en el sistema.

La unión de toda esta problemática estaba provocando una situación compleja en la que el usuario de negocio no podía utilizar el sistema actual con los requisitos exigidos, no es viable querer obtener el dato de ventas de una tienda con una demora de 2 días.

Adicionalmente al sobrecargado estado de uso del sistema, hay que añadir la investigación sobre nuevas funcionalidades para mejorar la frescura del dato. Se pretende llegar a conseguir que los usuarios de negocio tengan los datos más críticos con una frescura óptima, en el rango de los minutos.

Con este escenario era imposible satisfacer los requisitos utilizando el sistema tal y como estaba configurado.

Llegados a este punto se plantearon dos posibles soluciones:

1. Invertir en ampliar los recursos del sistema actual, servidores más potentes, almacenamiento más rápido, mejora en las comunicaciones.
2. Realizar una migración del entorno existente a una solución Hardware y Software específicamente diseñada para ser un Data Warehouse y que sirviese de base para abordar futuros proyectos con las nuevas tecnologías existentes en torno a la gestión de los datos.

La primera de las opciones, era la que menos impacto iba a causar, no era necesario modificar los procesos actuales y la inversión económica era inferior a corto y largo plazo. Pero mejorar el hardware de un sistema basado en procesos obsoletos tecnológicamente hablando no es una solución, el hardware que no está diseñado para ser un Data Warehouse por mucho que se mejore no puede competir en potencia de proceso frente a soluciones desarrolladas específicamente para ello.

A esta problemática hay que añadir que el entorno de ejecución de la lógica de negocio es muy cerrado y no permite la sencilla incursión de nuevas tecnologías; conexión a fuentes de datos como HADOOP, WebServices, Change Data Capture (captura de datos en tiempo real), paralelización de los procesos, son funcionalidades que o bien no se pueden obtener o su obtención no es sencilla.

La segunda opción suponía un cambio a nivel interno que engloba a muchos departamentos de la empresa. El coste de la adquisición de la solución es muy elevado a corto plazo y el proceso de migración de toda la lógica genera una gran inversión a largo plazo.

La solución elegida fue adquirir un sistema Data Warehouse específicamente desarrollado para esta función y comenzar a utilizar una herramienta de Extracción, Transformación y Carga de datos que permitiese adecuarse rápidamente a nuevas fuentes de datos y nuevos procesos de transformación.

Siendo la opción más costosa de las disponibles, era la mejor la solución ya que:

- Gracias a la potencia de un sistema específico se iba a poder reducir los tiempos de carga y consulta de datos.
- La utilización de una herramienta ETL iba a permitir la adecuación de los procesos actuales a las nuevas tecnologías existentes en el mercado así como a las metodologías de desarrollo de procesos ETL.
- La opción de tener una plataforma más moderna y potente, permite abordar funcionalidades requeridas imposibles de satisfacer en el sistema actual, por ejemplo obtener los datos con un tiempo inferior al actual.
- El sistema elegido sirve de base para abordar futuros proyectos de tecnologías "Big Data" y de Análisis predictivo de datos, ya que el sistema tiene soporte para ello.

Para la elección de un sistema Data Warehouse y una herramienta ETL se organizó una Prueba de Concepto (PoC) en la que los fabricantes elegidos implantaron su solución en el entorno del cliente y realizaron una serie de pruebas especificadas.

De entre las opciones disponibles en el mercado se eligió enfrentar a IBM y TERADATA ya que eran los mejores posicionados en las soluciones de Data Warehouse.

El PoC consistía en:

- Realizar la instalación física del sistema en el CPD.
- Ponerlo en marcha
- Realizar un volcado de toda la base de datos del sistema actual. Revisar tiempos de carga y ratios de compresión de datos
- Realizar pruebas de rendimiento de consultas complejas
- Realizar pruebas de concurrencia de usuarios en el sistema.
- Analizar el uso de la herramienta ETL en el proceso de la prueba

Resultados obtenidos de las pruebas reales, explicados por tareas:

Instalación en CPD

- IBM: Se realiza la instalación del sistema en 2 horas supervisada por un técnico especialista. El sistema IBM es más pesado por lo que es necesario reforzar el suelo del CPD.
- TERADATA: Se realiza la instalación del sistema en 3 horas ya que debido a la mayor altura del mismo es necesario desmontar parte del ascensor que da acceso al CPD.

Puesta en marcha del sistema

- IBM: La misma mañana de la instalación el sistema estaba disponible para su uso. Técnico especialista desplazado desde el extranjero que realiza la configuración inicial in situ.
- TERADATA: Fue necesario una conexión remota en días posteriores para terminar de configurar el sistema.

Volcado de datos del sistema actual, algunos datos de interés:

El sistema actual está desplegado en un iSeries server sobre una base de datos DB2, el número de tablas que componen la base de datos es de 326 con un tamaño total de ocupación en origen cercano a 4TB.

- IBM: Se utiliza la herramienta ETL de IBM, InfoSphere DataStage que dispone de conectores nativos para sistemas DB2 en iSeries. Se genera el DDL de todas las tablas y vistas del sistema utilizando la herramienta IBM Data Architect incluida en el empaquetado de la herramienta ETL. Se realiza el volcado de todos los datos del sistema en 4 días. La tabla de mayor número de registros, 2.907.078.847 con una ocupación en origen de 373GB se carga en 35horas y se obtiene una ocupación en destino de 65Gb con un ratio de compresión cercano al 5,7, se consigue un ratio máximo de hasta 11x en otras tablas.
- TERADATA: Se utilizan herramientas internas de carga de datos pero se encuentran con problemas a la hora de realizar conexiones al sistema iSeries, esto causa que se termine de realizar la carga completa del sistema en 15 días. Los tiempos de carga de las tablas con mayor volumen sobrepasan las 50 horas y los ratios de compresión son similares al sistema IBM.

Prueba de rendimiento de consultas complejas, se realizan pruebas de rendimiento sobre consultas preparadas para ello, una de la consulta más compleja nos puede dar una visión del resultado obtenido. Dicha consulta relaciona datos de artículos y sus ventas (fecha, ubicación, detalle de socio) de un año entero. Esta consulta en el sistema actual tiene un tiempo de ejecución de aproximadamente 24 minutos.

- IBM: Se ejecuta la sentencia y se obtiene un tiempo de ejecución de 8,4 segundos
- TERADATA: Se ejecuta la sentencia y se obtiene un tiempo de ejecución de 9,8 segundos

Con el resto de consultas analizadas se obtuvieron resultados similares entre los dos sistemas, el sistema IBM es ligeramente superior.

Prueba de concurrencia de usuarios en el sistema, se realiza una prueba de ejecución de una consulta no compleja pero con una concurrencia de usuarios elevada para analizar la respuesta del sistema.

- IBM:

Muestras	Media (ms)	Min (ms)	Max (ms)	%Error	Rendimiento/sec	Kb/sec	Media Bytes
600	1566	423	2509	0.0	238,9	52,83	226,40

Tabla 2 - Concurrencia de usuarios sistema IBM

- TERDATA:

Muestras	Media (ms)	Min (ms)	Max (ms)	%Error	Rendimiento/sec	Kb/sec	Media Bytes
600	1480	400	2015	0.0	250,7	53,05	227,04

Tabla 3- Concurrencia de usuarios sistema TERADATA

Como se puede comprobar el rendimiento del sistema TERADATA es ligeramente superior para concurrencia de cargas.

Análisis de la herramienta ETL

En este aspecto hubo un claro vencedor ya que IBM proporciona una herramienta ETL plenamente integrada con los sistemas actuales de origen y con el Data Warehouse evaluado. TERADATA no dispone de una herramienta ETL al uso, sino de pequeñas herramientas internas para la carga de datos que no aportan toda la funcionalidad requerida.

Decisión

Después del análisis de las pruebas, los dos sistemas podían satisfacer las necesidades de procesamiento de datos y mejoraban al sistema actual con creces. En la herramienta ETL como ya hemos comentado desde un principio se tuvo claro que la seleccionada era InfoSphere DataStage ya que se había comprobado su plena integración con el ecosistema IBM ya instalado y satisfacía todas las necesidades de carga de datos actuales y futuras.

En cuanto al sistema Data Warehouse al final, se apostó por implantar una solución de continuidad en el entorno del departamento de tecnologías de información y se seleccionó la solución de IBM PureData System for Analytics. En este caso la elección fue una decisión no solo a nivel tecnológico sino a nivel administrativo y de línea de negocio, ya que las diferencias entre los dos sistemas al puro nivel de rendimiento no son tan significativas.

3 SOLUCIÓN DATAWAREHOUSE + ETL, EN CLIENTE DEL SECTOR DISTRIBUCION ALIMENTARIA

3.1 ARQUITECTURA DATAWAREHOUSE

Como ya se ha comentado en puntos anteriores, la solución elegida como sistema Data Warehouse es IBM PureData System for Analytics, parte de la siguiente descripción del producto ya se ha expuesto en el apartado 2.2.1.1 IBM PURE DATA SYSTEM FOR ANALYTICS.

IBM PureData for Analytics es un sistema de alto rendimiento, escalable, de ejecución asimétrica masivamente paralela (AMPP), que proporciona una plataforma analítica capaz de gestionar volúmenes de datos enormes.

Es un dispositivo ofrecido en formato appliance, es decir, con todos sus componentes ya integrados, instalados y configurados de fábrica, integra en su arquitectura una base de datos de altas prestaciones, el hardware servidor preciso para ejecutar el software embebido y el almacenamiento necesario. Este sistema, basado en tecnología de Netezza, ha sido diseñado específicamente para ejecutar cargas analíticas muy complejas sobre volúmenes ingentes de datos, pero de una manera muy sencilla, haciendo que los costes de mantenimiento y operación sean mucho menores que con otros sistemas existentes.

La base del diseño de PureData for Analytics es la de tratar de eliminar en la medida de lo posible, el movimiento de datos a lo largo del sistema, llevando la ejecución de los procesos analíticos a donde residen los datos, en lugar de realizar el camino contrario.

Gracias a esto, el rendimiento ofrecido por PureData for Analytics permite realizar análisis sobre los datos que hasta ahora no eran posibles de realizar, al tiempo que se mejoran los rendimientos de las tareas existentes. Es una solución construida específicamente para resolver este tipo de cargas analíticas, basado en estándares de gestión de sistemas de Data Warehouse, integrando en un solo producto el software de base de datos, los servidores hardware, el almacenamiento y las capacidades analíticas avanzadas, es decir, se proporciona una solución integrada y completa, preinstalada y preconfigurada, que no requiere *tuning* adicional para conseguir el rendimiento ofrecido por el sistema.

3.1.1 HARDWARE

El sistema específico propuesto en la solución es el modelo N2001-005, que dispone de las siguientes características

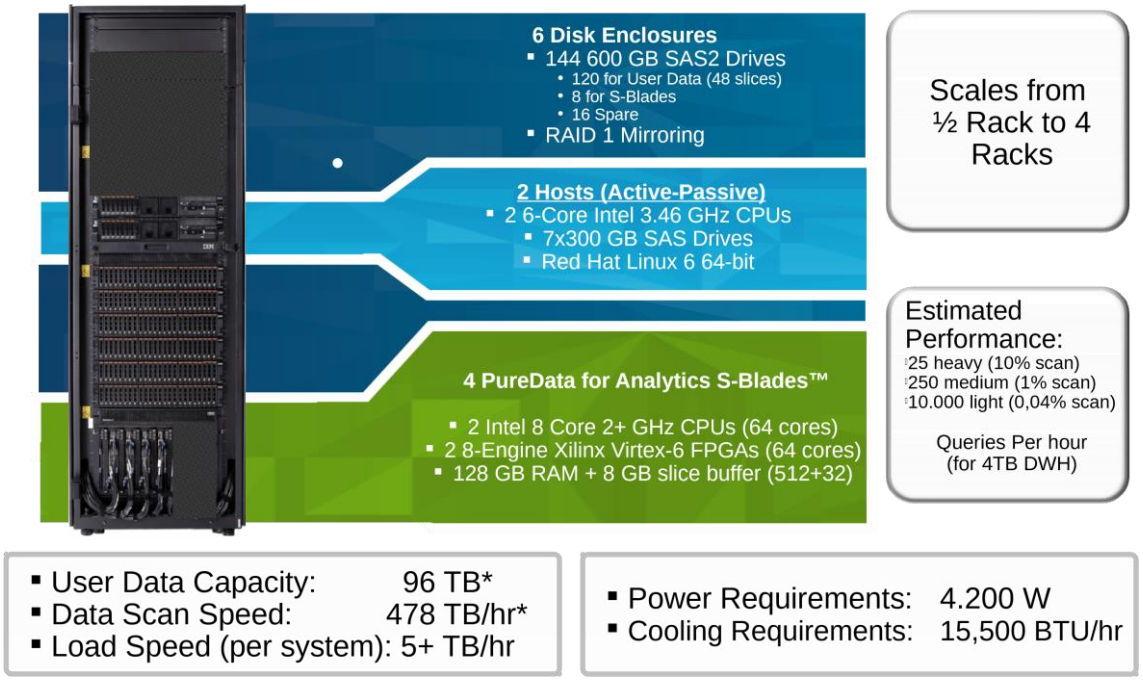

Ilustración 9 - PureData System for Analytics

3.1.1.1 Almacenamiento

El sistema dispone de 6 Bandejas de discos con tecnología SAS que proporcionan 120 dataslices (unidades mínimas de almacenamiento) y una capacidad neta del sistema de 23TB raw que equivalen a unos 96TB de uso de datos con una compresión aproximadamente de 4X.

Cada disco está protegido mediante data mirroring en RAID-1. De este modo, los discos se dividen en tres particiones principales: datos de usuario (Primary), mirror (espejo o copia) de la partición de datos de otro disco y espacio temporal. El espacio temporal también se encuentra replicado para asegurar que ante el fallo de un disco, las consultas que se estén realizando en ese momento no se interrumpirán nunca.

Además, se incluyen una serie de discos de repuesto en modo hot-spare, es decir, que ante el fallo de un disco, de forma automática el sistema activa uno de los discos de repuesto, copia los datos del disco que ha fallado (desde la copia de seguridad mirror de otro disco), y activa el disco de repuesto para que actúe en sustitución del disco que ha fallado.

Adicionalmente, generará una alarma para advertir al administrador de que debe sustituir el disco que falló por otro disco nuevo.

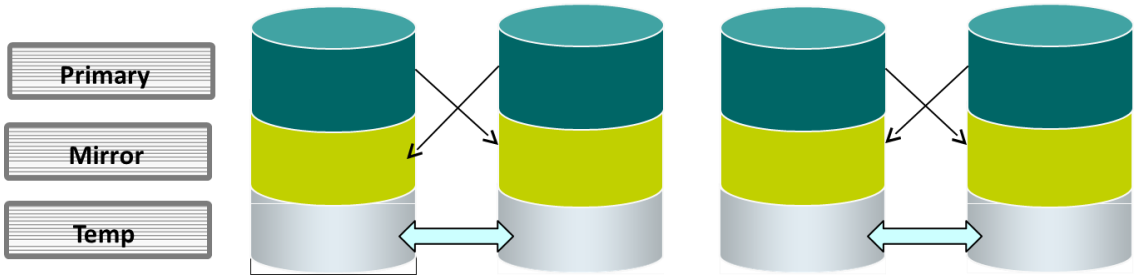

Ilustración 10 - Estrategia de protección de discos

3.1.1.2 Networkiong

En cuanto a la red de interconexión interna de 10GbE, está completamente redundada, con routers y switches duplicados. Si un switch de la red o router fallara, la red redundante se convertiría en la red activa.

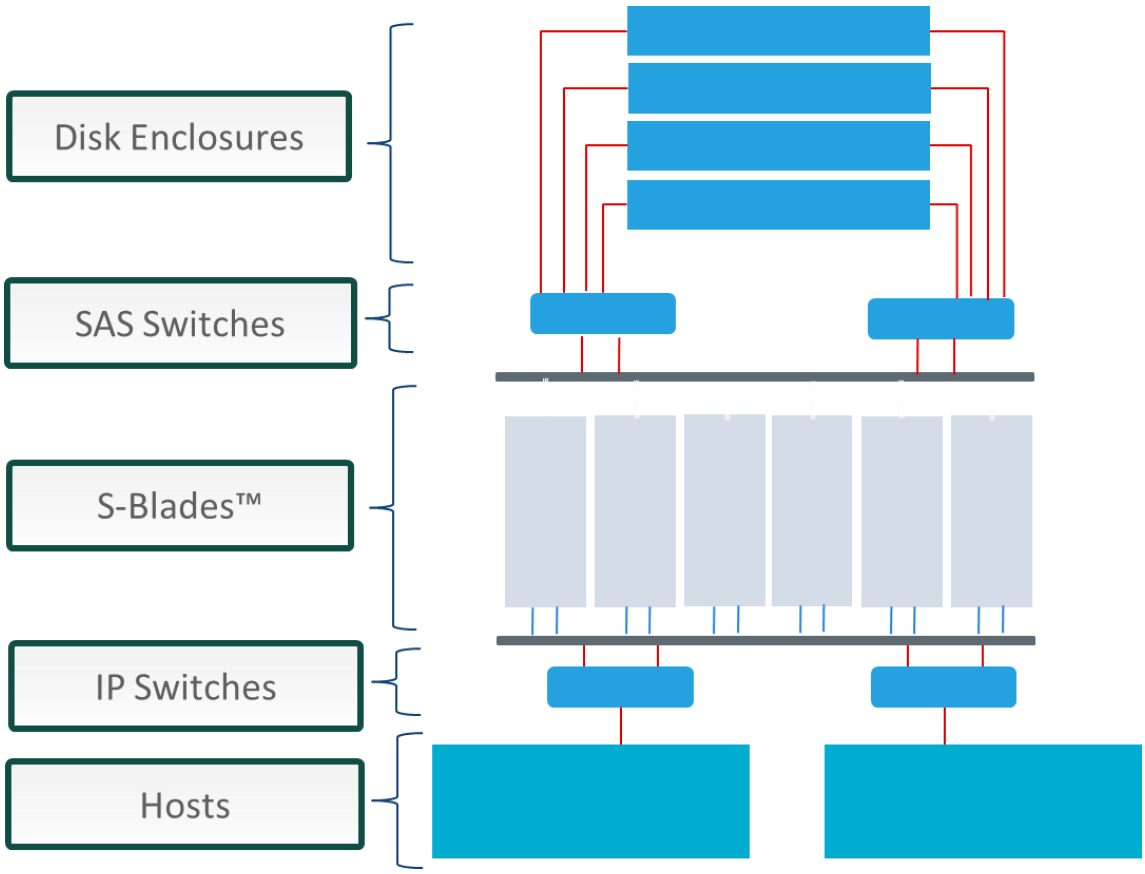

Ilustración 11 - Mapa de Red del sistema

3.1.1.3 Nodos de gestión en alta disponibilidad

Los nodos de gestión del sistema están compuesto por dos servidores xSeries IBM configurados en un clúster en alta disponibilidad Linux (Linux-HA) con las siguientes características:

- Dos *hosts* en configuración *clúster*:
  - Activo (Primario)
  - Standby (Secundario)
- Mecanismo de replicación de bloques entre los discos de los dos *hosts* para asegurar la copia de los datos:
  - Distributed Replicated Block

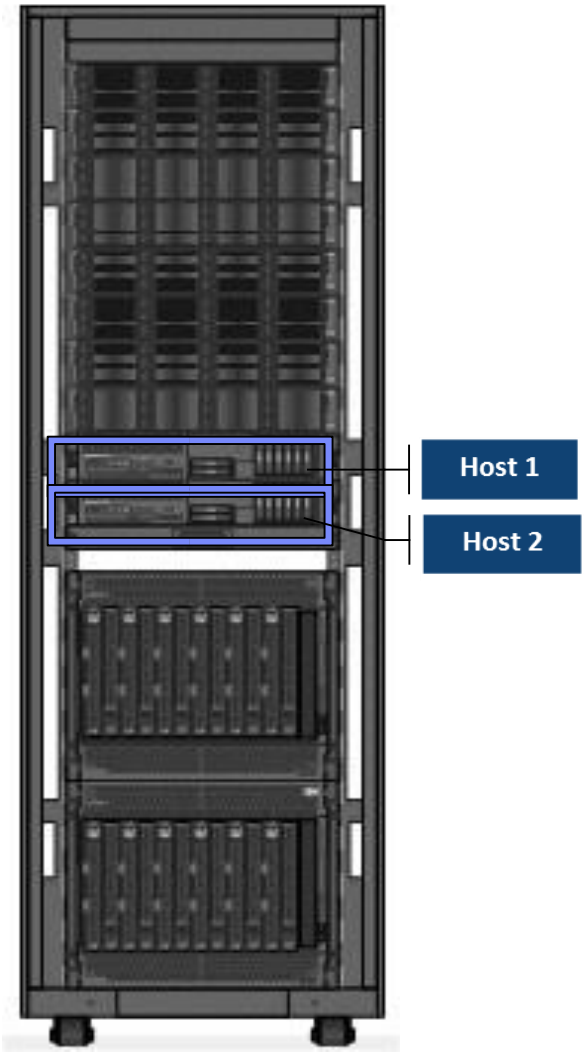

Ilustración 12 - Esquema Rack PureData

3.1.1.4 Módulos de procesamiento: S-BLADES

El Core de la solución, la parte que realmente le otorga la potencia, está formada por módulos de procesamiento llamados S-BLADES que proporcionan una aceleración hardware en el procesamiento de consultas SQL.

Cada módulo (o cuchilla) S-BLADE está formada por dos componentes que funcionan conjuntamente.

Mediante el uso de la tecnología side-car de IBM, a un blade (cuchilla) estándar basada en CPU Intel se le acopla una tarjeta aceleradora de consultas de la base de datos, Netezza DB Accelerator, de forma que ambas tarjetas se encuentran físicamente unidas en el mismo elemento hardware, beneficiándose de su proximidad para trabajar de forma conjunta sin tener que recurrir a mover los datos a través de la red de comunicaciones interna del sistema.

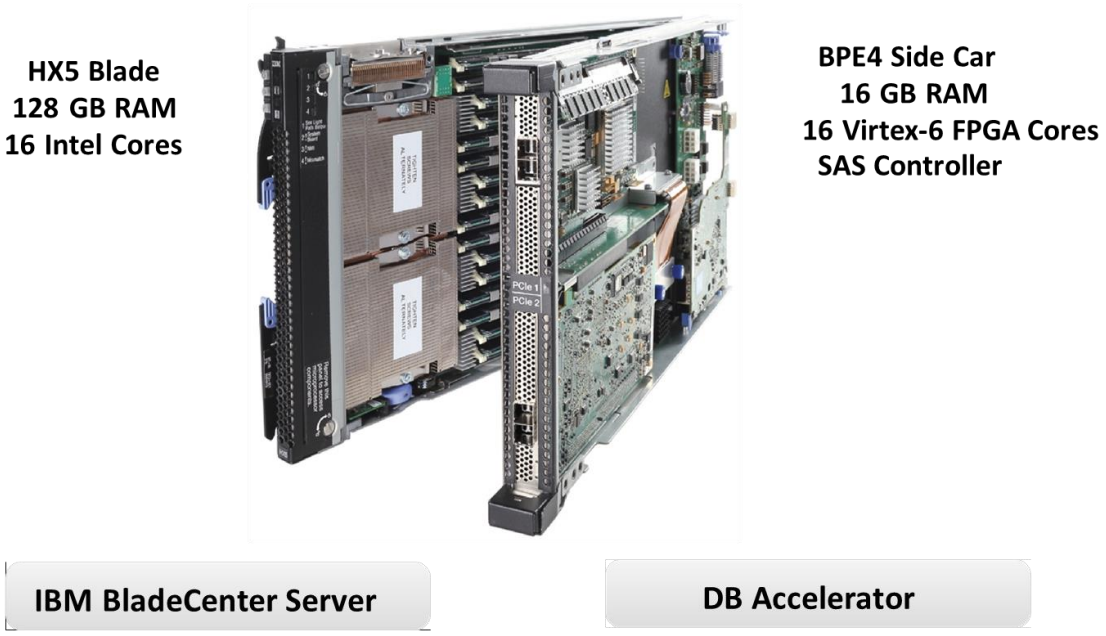

Ilustración 13 - BladeCenter + DBA Accelerator

De esta forma, utilizando componentes hardware estándar, de bajo coste, se consigue el mismo rendimiento que si se construyera un hardware específico para la realización de las consultas a la base de datos, pero a un coste de mantenimiento y operación mucho menor.

El sistema DB Accelerator procesa los datos por medio de FPGAs. El FPGA (Field Programmable Gate Array) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada 'in situ' mediante un lenguaje de descripción especializado.

Es decir, en lenguaje llano, es un microprocesador que se puede reconfigurar en cada momento para realizar cualquier tipo de función lógica.

En lo que aplica a PureData for Analytics, es como si cada una de las queries que se realizan sobre el sistema contara con un microprocesador creado específicamente para realizar la consulta que se ha enviado al sistema en un momento dado, ya que es el propio sistema el que se encarga de realizar esta reconfiguración con cada query que se ejecuta.

Además de hacerlo de la forma más eficiente posible, hay que sumar la velocidad que supone que esta reprogramación se realiza sobre un dispositivo hardware, frente al coste que tendría realizar esta misma consulta con un mecanismo exclusivamente software.

En el sistema las FPGAs se utilizan como aceleradores en los procesos de lectura de los discos, para tareas de descompresión y para el filtrado de los datos, garantizando que no existen cuellos de botella en el I/O.

Ilustración 14 - Netezza DB Accelerator

3.1.1.4.1 S-Blade Tolerancia a fallos

En lo que se refiere a los nodos de computación en sí, las S-Blades, también cuentan con redundancia y tolerancia a fallos, e incluso se cuenta con un nodo adicional, en modo hot-spare, que se activa automáticamente en caso de fallo de una de las SBlades. Esta característica, permite que ante el hipotético fallo de una de las S-Blades, el sistema se recupere automáticamente sin perder rendimiento en absoluto, los discos que controla la S-Blade que no funciona se reasignan automáticamente al resto. Ante esta eventualidad, el administrador recibiría el aviso del fallo de la S-Blade, pero los usuarios no notarían ningún cambio en el rendimiento del sistema.

Aún en el muy raro supuesto de que se produjera el fallo de una segunda S-Blade en el mismo sistema, éste no se detendría en ningún caso, aunque se produciría una pérdida de rendimiento, ya que las S-Blades restantes se harían cargo de resolver la ejecución de las queries lanzadas contra el sistema.

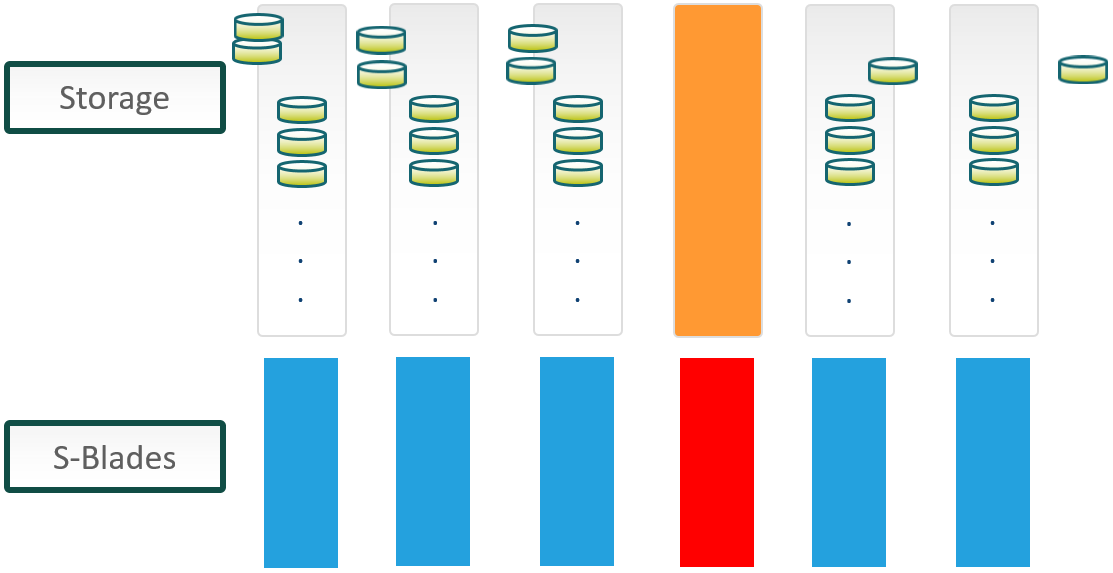

Ilustración 15 - Comportamiento del sistema ante la caída de un S-Blade

3.1.2 SOFTWARE

El appliance IBM Pure Data System for Analytics es una conjunción de software y hardware en una sola máquina. De esta conjunción de software y hardware obtenemos una serie de soluciones que proporcionan al sistema su gran capacidad de procesamiento de datos.

3.1.2.1 Asymmetric Massively Parallel Processing

IBM PureData for Analytics se construye mediante una arquitectura AMPP para la ejecución de los procesos dentro de la base de datos, basada en el concepto "shared-nothing" que permite maximizar el rendimiento de la mencionada arquitectura AMPP.

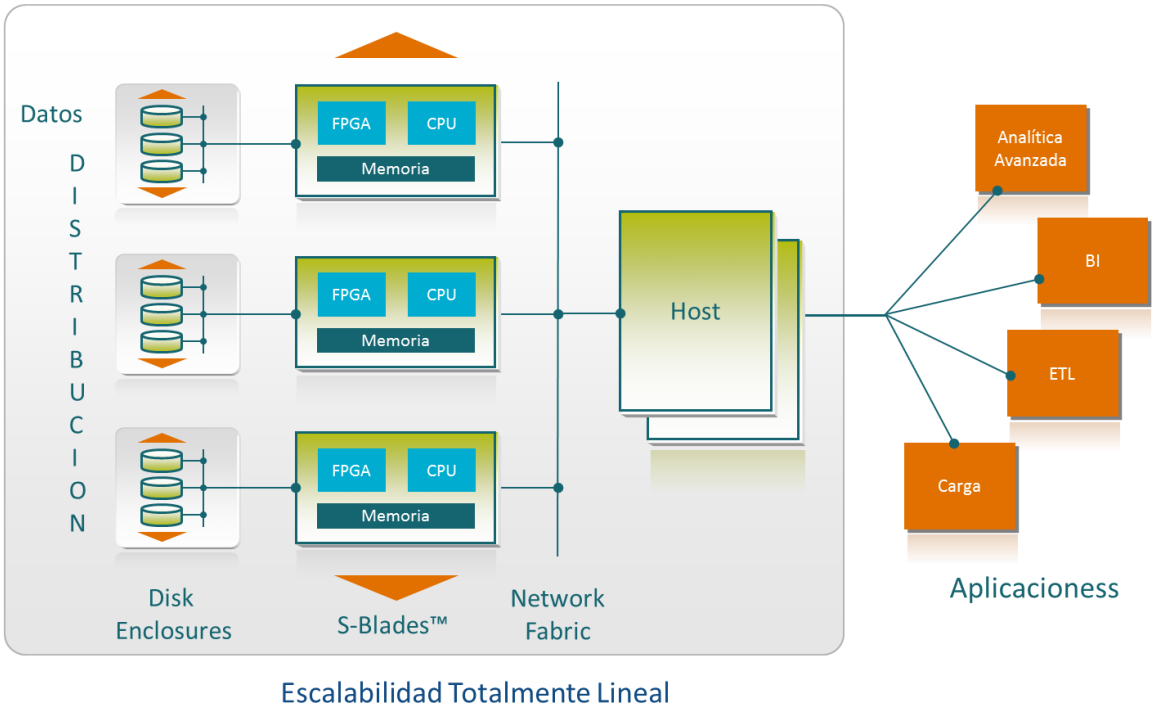

Ilustración 16 - Sistema de Procesamiento Paralelo Masivo de Netezza (I)

Esta arquitectura está construida en torno a 2 capas diferenciadas:

La primera capa es la compuesta por los dos hosts que constituyen el interfaz de acceso de los usuarios del sistema y que sirven para ocultar a los citados usuarios finales la complejidad de ejecución que subyace a la mencionada arquitectura AMPP.

Las queries ejecutadas en el sistema, sentencias SQL por lo general, llegan a esta capa a través de la red desde aplicaciones externas. Hay que incidir en el hecho de que se trata

únicamente de una capa de acceso, muy ligera, y que ni los datos residen en esta capa, ni el procesamiento del sistema se ejecuta en ningún caso dentro de esta capa.

Este host recibe la query, se compila y se genera un plan de acceso, que es distribuido a través de la red interna del sistema hacía la 2ª capa, que es la capa AMPP.

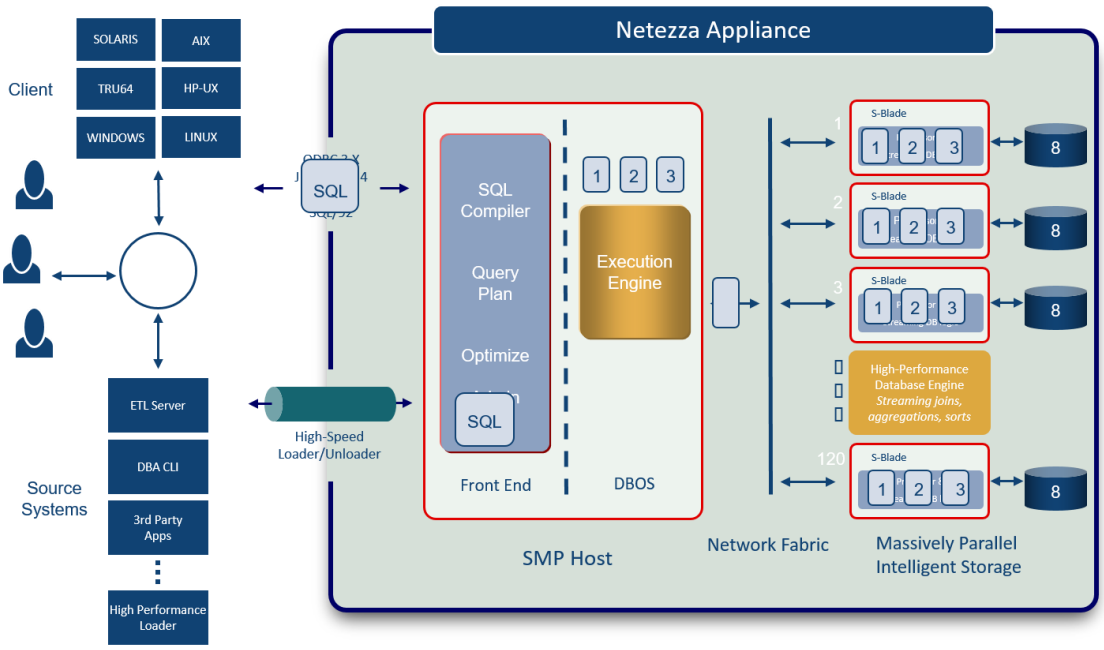

Ilustración 17 - Sistema de Procesamiento Paralelo Masivo de Netezza (II)

La segunda capa consiste en IBM Blades estándar con una tarjeta aceleradora "side car" donde residen las CPUs FPGA (Field Programmable Gate Array). A su vez, cada uno de estos Blades tienen asignados un grupo de discos, de forma que se componen los nodos de computación a razón de un disco duro asociado a un core de CPU Intel y a una CPU de procesador FPGA. Esta asignación consigue un sistema muy equilibrado y que permite realizar el citado procesamiento masivo en paralelo que se describe aquí.

En esta capa AMPP cada nodo (HDD+Intel+FPGA) procesa una parte de las tablas indicadas en la query, utilizando una estrategia clásica de "Divide y Vencerás". Tal y como se ha mencionado en este mismo apartado, la arquitectura del sistema también implementa el principio "shared-nothing" en el almacenamiento de los datos. Es decir, la información que se almacena en el sistema se divide entre los diferentes discos de los nodos de computación del sistema, en conjuntos totalmente disjuntos entre sí, por lo que pueden ser tratados al mismo tiempo y en paralelo, potenciando la eficiencia de esta arquitectura AMPP y de la citada estrategia "divide y vencerás".

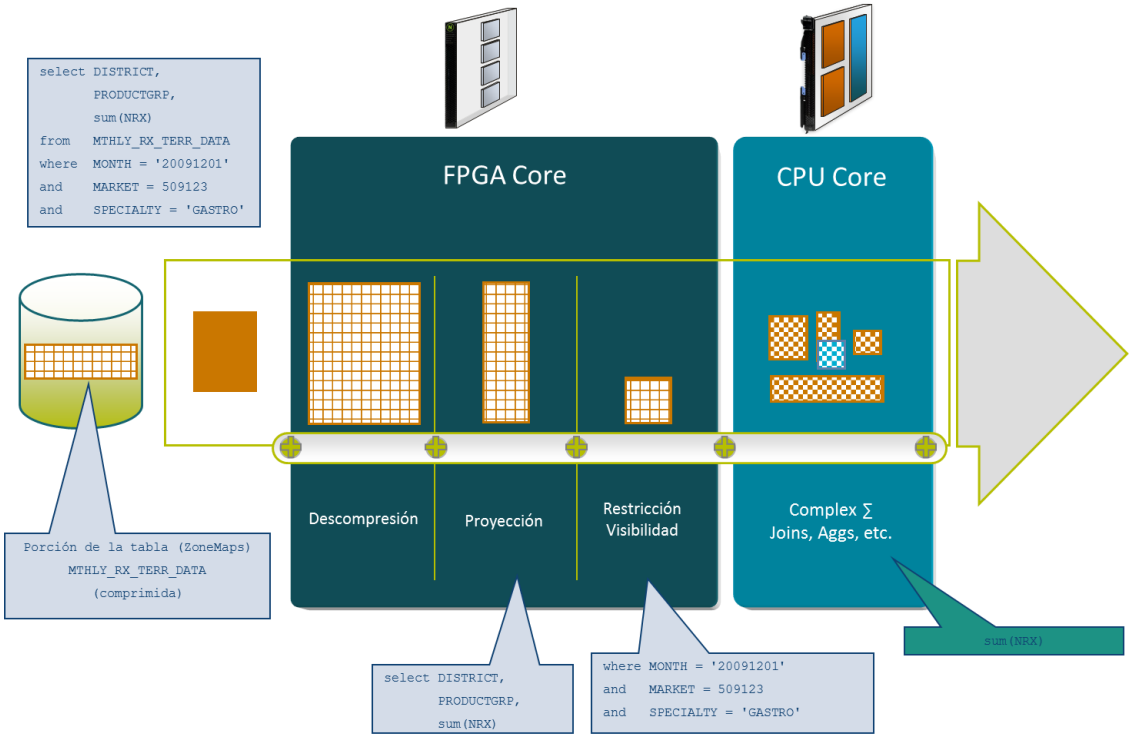

Ilustración 18 - Proceso de ejecución de un query en Netezza

Así, la potencia del PureData System for Analytics es su habilidad de trabajar a la velocidad "física" máxima que permite el sistema. De este modo, desaparecen las limitaciones de E/S, no se generan cuellos de botella y se logra procesar los datos a la velocidad del disco.

Las conexiones con aplicaciones externas se realizan a través de SQL, ODBC y JDBC.

La carga de datos se realiza con utilidades de carga rápida.

3.1.2.2 Compresión de los datos

Todos los datos que se almacenan dentro de PureData System for Analytics, lo hacen de forma comprimida. Mediante un algoritmo patentado de IBM, todos los tipos de datos que se almacenan en PureData se comprimen de forma automática, siguiendo estrategias diferentes según el tipo de dato en el momento de la carga de los mismos, para garantizar que se alcanza la máxima compresión posible.

Esto hace que la capacidad de almacenamiento neta del sistema aumente considerablemente: el ratio mínimo de compresión que se garantiza es de 4 veces el tamaño original del dato de media, siendo frecuente alcanzar hasta 10 veces, con hasta 32 veces en ratios de compresión, dependiendo de la naturaleza de los datos comprimidos.

El algoritmo de compresión es un algoritmo de compresión columnar, lo que le permite alcanzar los niveles de compresión anteriormente mencionados, pero en el almacenamiento se guarda la referencia de las filas que componen los datos de la base de datos que se está gestionando. Esta referencia o almacenamiento basado en filas, es muy importante para garantizar el rendimiento global del sistema.

Otro punto importante que hay que destacar es que el mecanismo de compresión que se incluye en PureData for Analytics no requiere ninguna intervención por parte del usuario final. Es decir, el mecanismo de compresión está siempre activo en el sistema y no es necesario que el administrador ni ninguno de los usuarios realicen ninguna tarea de configuración ni de *tuning* sobre el sistema para el funcionamiento del mecanismo mencionado de compresión.

3.1.2.3 Tecnología ZoneMaps: No hacen falta índices

Una de las principales características del sistema, es que no es necesario configurar índices ni ningún otro artificio para acelerar el acceso a los datos almacenados en la base de datos.

La razón por la que no es necesario implementar índices de ningún tipo, es la tecnología ZoneMaps que implementa PureData for Analytics. Esta tecnología consiste en dividir las unidades de almacenamiento de la base de datos en pequeños bloques de 128Kb.

Para estos bloques, se guarda el valor máximo y mínimo de los valores de las columnas de la base de datos soportados. De este modo, cuando se realiza una consulta a la base de datos, no es necesario leer toda la base de datos, sólo se transmiten hacia la parte de análisis –la parte del nodo compuesto por FPGA+CPU+Memoria RAM- aquellas zonas que son candidatas a contener los datos relevantes para la query.

Con esta tecnología, recordemos que se trata de zonas de apenas 128Kb, se reduce drásticamente la entrada/salida en el sistema, ya que no se ha de leer toda la base de datos, sino sólo aquellas zonas, de pequeño tamaño y que además se almacenan comprimidas, que son realmente relevantes para la consulta.

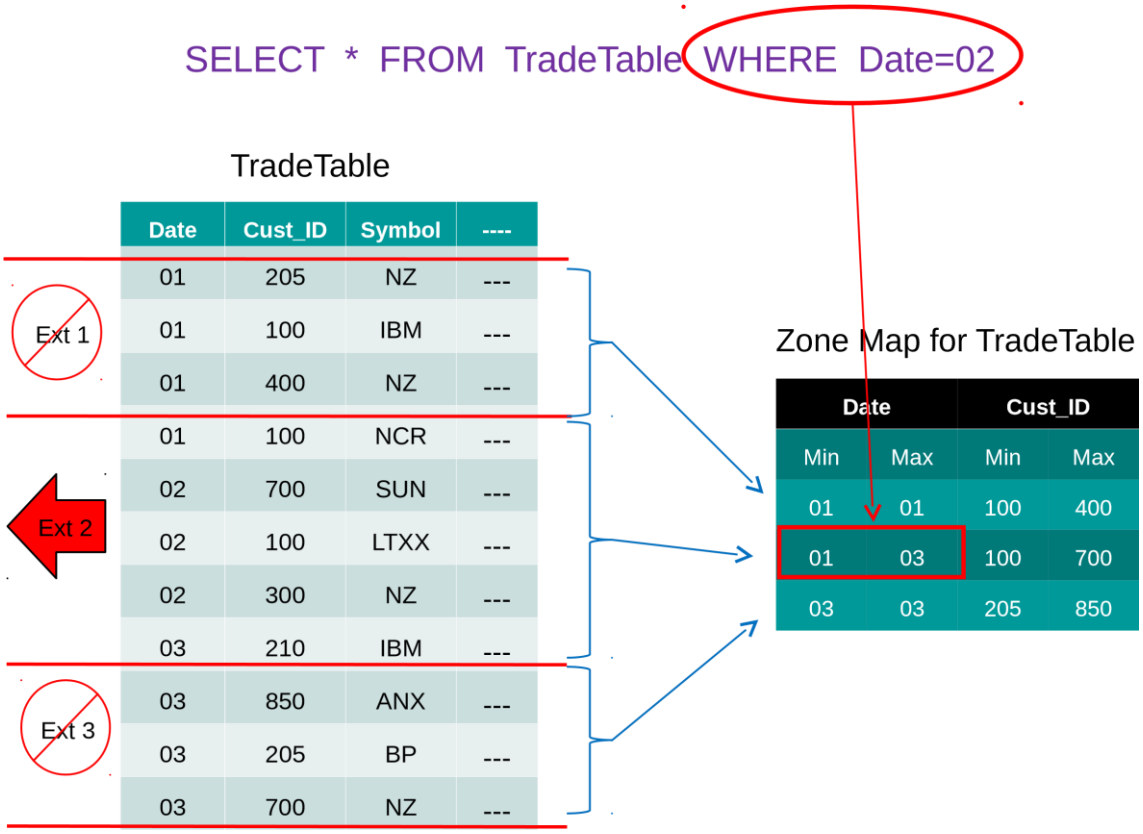

Ilustración 19 - Uso de ZoneMaps en Netezza

Otro punto a destacar de esta tecnología es que su uso es "gratuito", es decir, no supone ninguna penalización en el rendimiento del sistema en el momento de realizar las escrituras sobre el mismo. Los ZoneMaps se actualizan automáticamente ante cualquier escritura, borrado o modificación de la base de datos, sin suponer estas actualizaciones ninguna alteración en cuanto al rendimiento esperado del sistema.

Una característica adicional de los ZoneMaps es que no necesitan de la intervención del administrador en ningún caso. No sólo se actualizan de forma totalmente automática, sino que el administrador no tiene que preocuparse en absoluto de su mantenimiento, al contrario de lo que sucede con la creación de índices u otros mecanismos de aceleración de las consultas, con lo que el coste de propiedad y mantenimiento del sistema disminuye notablemente.

3.1.2.4 Simplicidad

A la hora de diseñar el IBM PureData for Analytics, se ha tenido muy en cuenta que debe ser un sistema muy sencillo de gestionar y operar por el usuario final. Esta simplicidad de uso se logra en gran parte gracias a la ausencia de índices, particiones, gestión del *mirroring*, de la alta disponibilidad, etc., que aseguran un muy bajo coste de operación y mantenimiento y garantizan que los nuevos desarrollos se pueden poner en producción en mucho menor tiempo que con una base de datos tradicional, redundando en un menor coste de explotación de la plataforma.

En PureData for Analytics NO es necesario realizar las siguientes tareas:

- NO hay que crear Índices (¡no existen!)
- NO hay que gestionar el particionado del sistema (el sistema lo hace automáticamente)
- NO hay que configurar el dimensionamiento de dbspace/tablespace
- NO hay que configurar o dimensionar redo/physical log
- NO hay que configurar o dimensionar journaling/logical log
- NO hay que configurar ni dimensionar el tamaño de página/bloque para las tablas
- NO hay que configurar ni dimensionar el extent para las tablas
- NO hay que ubicar espacio para temporal y monitorización
- NO hay que configurar o decidir sobre el nivel de RAID en los dbspaces
- NO hay que crear volúmenes lógicos
- NO hay que realizar mantenimiento del nivel de parcheado del sistema operativo
- NO hay que hacer mantenimiento de la integración de la base de datos y el sistema operativo
- NO hay que tener reuniones entre diferentes equipos para la configuración de los servidores/redes/almacenamiento
- NO hay software que instalar (es un *appliance*, viene instalado y configurado de fábrica)
- SOLO UNA simple estrategia de distribución de los datos: HASHING

En PureData for Analytics las tareas que se realizan para la configuración y puesta en marcha de un Data Warehouse son las siguientes:

- Determinar si los datos los debe distribuir el sistema en función de un campo determinado, o dejar que sea el sistema el que lo haga todo de forma automática y equilibrada (round-robin/hash)
- Crear las tablas de la base de datos
- Cargar los datos
- Actualizar las estadísticas cuando sea necesario

3.2 ARQUITECTURA ETL

Como herramienta de Extracción, Transformación y Carga de datos se ha decidido por el módulo IBM InfoSphere DataStage que pertenece a la familia de IBM InfoSphere Information Server.

InfoSphere Information Server proporciona una plataforma única para la integración de la información. Los componentes de la suite se combinan para crear una base unificada para arquitecturas de información de la empresa, capaz de escalar para cumplir con cualquier requisito de volumen de información.

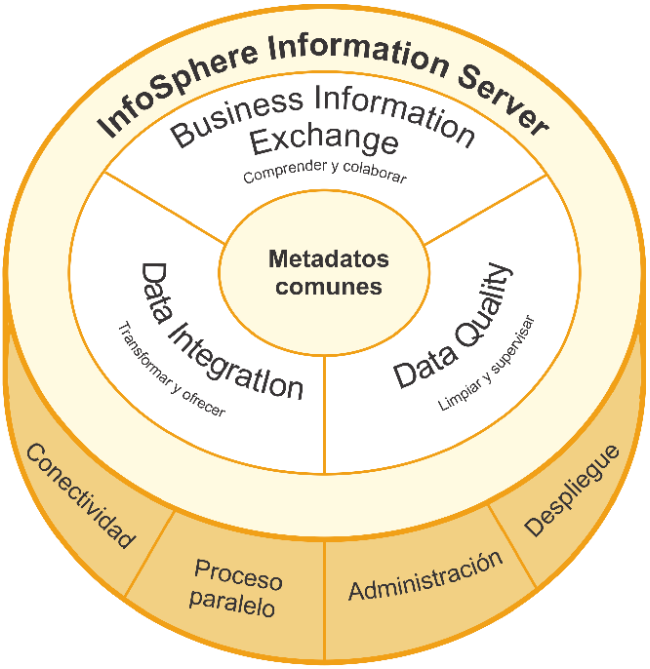

Ilustración 20 - Familia InfoSphere Information Server

3.2.1 HARDWARE

La plataforma elegida para realizar la instalación del producto es un servidor IBM POWER8 modelo 8286-42A IBM Power S824 con un sistema operativo IBM AIX v7.1. Se ha decidido utilizar esta plataforma ya que el cliente tiene su core de negocio desplegado sobre sistemas IBM POWER con sistema operativo i5/OS y dispone de recursos para la gestión de un nuevo sistema de estas características.

En cuanto al sistema operativo, AIX es un sistema UNIX y el cliente dispone de recursos con conocimientos en él.

3.2.1.1 Servidor POWER8 8286-42A

El sistema elegido es un modelo 8286-42A IBM Power S824 que dispone de las siguientes características:

Ilustración 21 - 8286-42A IBM Power S824

Configuración del Sistema	8286-42A
Microprocesador	Un procesador 8-core 4.15 GHz POWER8
Level 2 (L2) cache	512 KB L2 cache por procesador
Level 3 (L3) cache	8 MB L3 cache per core
Level 4 (L4) cache	16 MB per DIMM
Memoria	512 GB
Ancho de banda memoria-procesador	192 GBps por socket
Backplane almacenamiento	12 bahías para Hard Disk Drive (HDD)/Solid State Disk (SSD)
Bahia multimedia	1 slim DVD
Controlador SAS integrada	Estandar RAID 0,5,6,10
Slots Adaptadores	8 PCIe, 11 PCIe Gen3, 2 CAPI
I/O Ancho de banda	96 GBps por socket
Dimensiones	427.5 Ancho x 173 Alto x 750.5 Largo mm
Fuente de alimentación	100 V to 240 V

Tabla 4 - Configuración del Sistema 8286-42A

3.2.1.2 Almacenamiento

El sistema no dispone de disco duro interno ya que tanto los discos de sistema operativo como los discos de datos están ubicado en una cabina de almacenamiento externo IBM V7000 perteneciente al cliente.

Ilustración 22 - IBM v7000

La configuración de almacenamiento implementada es la siguiente:

Dentro del espacio disponible en la cabina se han generado 5 LUNs para dividir el almacenamiento del que disponemos.

A nivel de configuración de sistema operativo AIX tenemos disponibles 5 hdisks asociados a las LUNs:

Utilizando la potencia de configuración de almacenamiento de AIX se han generado dos Volume Group principales, rootvg y datavg.

Un *Volume Group* de AIX es una conjunto de almacenamiento compuesto por 1 o varios Physical volumes. Este conjunto pude abarcar varios discos físicos.

Un *Physical Volume* es un volumen físico, generalmente asociado a un 1 disco físico o a un hdisk en terminología AIX que es un disco lógico formado por una asociación de discos.

Un Logical Volume se trata de un espacio de disco destinado a contener un sistema de ficheros, es un equivalente a la partición de disco clásica.

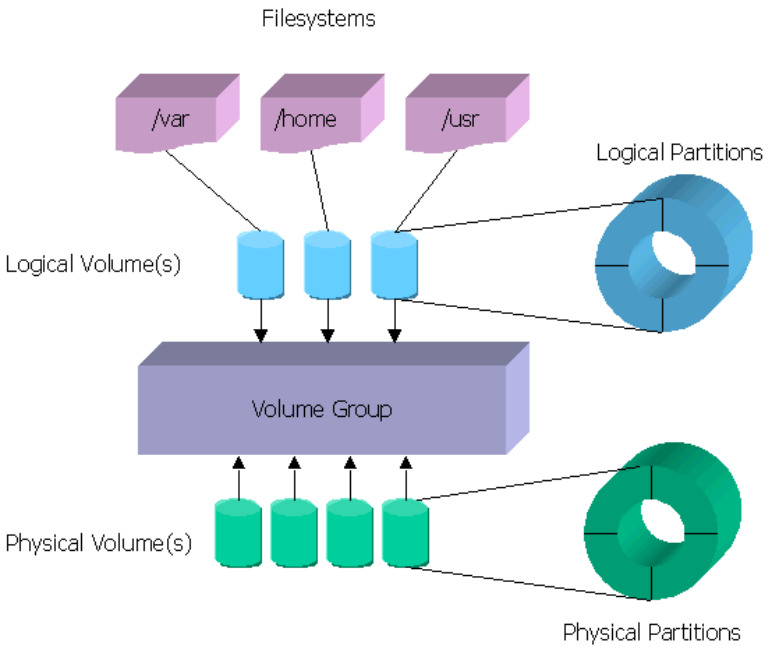

Ilustración 23 - Estructura del Logical Volume Manager [Davis Mendoza Paco]

El hdisk0 alberga el volume group rootvg, donde realizaremos la instalación de los binarios del producto IBM InfoSphere DataStage.

Los hdisk1, hdisk2, hdisk3, hdisk4 componen el volume group datavg donde residirán los sistemas de ficheros necesario para el funcionamiento paralelo del producto.

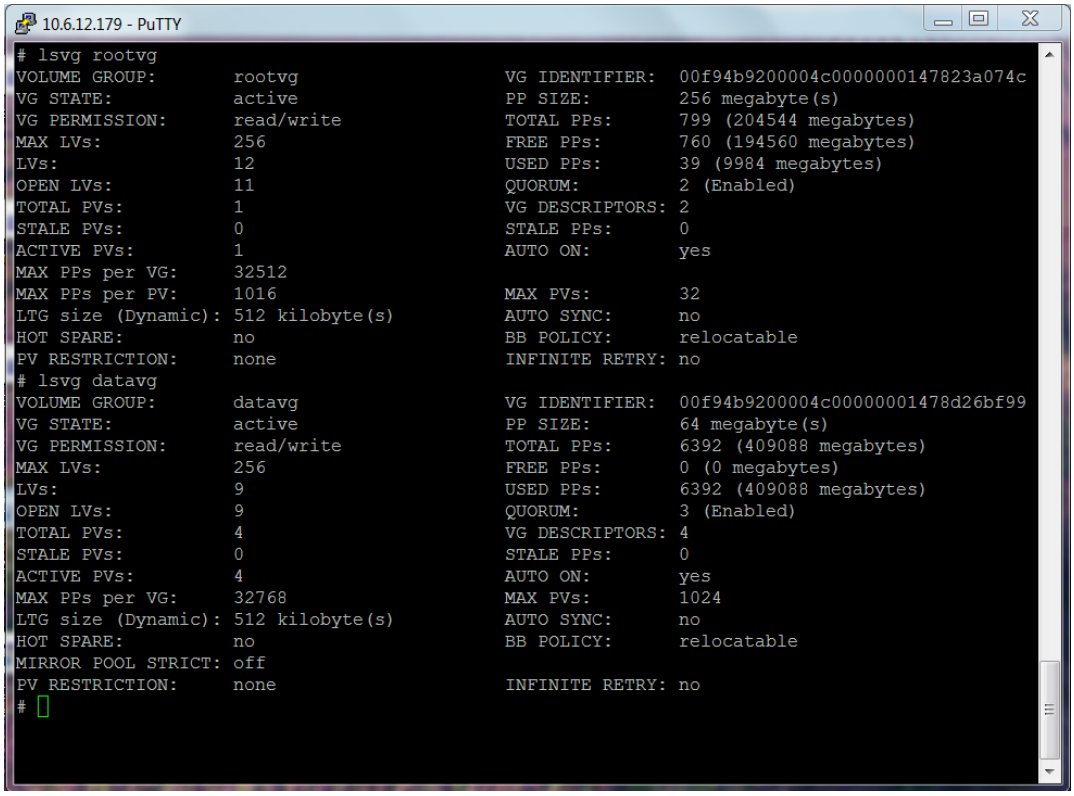

Ilustración 24 - Resumen de configuración de VolumeGroup

Una vez generado el Volume group datavg, donde van a residir los sistemas de ficheros que utiliza IBM InfoSphere DataStage, es necesario crear, un logical volume por file system asociado a cada hdisk de los que disponemos; es decir crearemos un disk1lv de tipo jfs2 con el punto de montaje /fs1/ds/disk. Adicionalmente crearemos un scratch1lv del tipo jfs2 con el punto de montaje /fs1/ds/scratch.

Repetiremos este proceso hasta crear 4 LV de cada tipo. También es necesario crear un LV para logs, cuyo nombre es logdatavglv de tipo jfs2log y 1 PPs de tamaño. El tamaño de LPs asignado a al resto de los LV se extrae a partir de la división del tamaño global del VG (Volume group) entre el número de LV que vamos a crear.

En nuestro caso disponemos de 6392 PPs, y vamos crear 8 LV, 2 por cada hdisk. Por lo que cada LV tendrá un tamaño de 799 PPs a excepción del scratch4lv que tiene un tamaño de 798 debido al PPs utilizado por el LV de log.

```
datavg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
disk1lv             jfs2       799     799     1    open/syncd    /fs1/ds/disk
disk2lv             jfs2       799     799     1    open/syncd    /fs2/ds/disk
disk3lv             jfs2       799     799     1    open/syncd    /fs3/ds/disk
disk4lv             jfs2       799     799     1    open/syncd    /fs4/ds/disk
scratch1lv          jfs2       799     799     1    open/syncd    /fs1/ds/scratch
scratch2lv          jfs2       799     799     1    open/syncd    /fs2/ds/scratch
scratch3lv          jfs2       799     799     1    open/syncd    /fs3/ds/scratch
scratch4lv          jfs2       798     798     1    open/syncd    /fs4/ds/scratch
logdatavglv         jfs2log    1       1       1    open/syncd    N/A
```

Ilustración 25 - Resumen de logical volumes del volume group datavg

El siguiente paso ha sido generar los puntos de montaje de cada LV. Esta acción se ejecuta automáticamente lanzando el comando: mount –all.

Para ver que realmente se han generado los path con el tamaño deseado, ejecutamos el comando: df -g

```
# df -g
Filesystem    GB blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           0.50      0.32   37%    10053    12% /
/dev/hd2           2.25      0.37   84%    42051    32% /usr
/dev/hd9var        0.50      0.22   57%     6174    11% /var
/dev/hd3           0.25      0.25    2%       45     1% /tmp
/dev/hd1           0.25      0.25    1%        5     1% /home
/dev/hd11admin     0.25      0.25    1%        5     1% /admin
/proc                 -         -    -         -     -  /proc
/dev/hd10opt       0.50      0.31   38%     7037     9% /opt
/dev/livedump      0.25      0.25    1%        4     1% /var/adm/ras/livedump
/dev/disk1lv      49.94     49.93    1%        4     1% /fs1/ds/disk
/dev/disk2lv      49.94     49.93    1%        4     1% /fs2/ds/disk
/dev/disk3lv      49.94     49.93    1%        4     1% /fs3/ds/disk
/dev/disk4lv      49.94     49.93    1%        4     1% /fs4/ds/disk
/dev/scratch1lv     49.94     49.93    1%        4     1% /fs1/ds/scratch
/dev/scratch2lv     49.94     49.93    1%        4     1% /fs2/ds/scratch
/dev/scratch3lv     49.94     49.93    1%        4     1% /fs3/ds/scratch
/dev/scratch4lv     49.88     49.87    1%        4     1% /fs4/ds/scratch
#
```

Ilustración 26 - Resumen de los FileSystems creados

3.2.1.3 Networking

El servidor elegido dispone de una tarjeta de red compuesta por dos puertos 10 Gb de fibra y dos puertos 1 Gb Ethernet.

Se han configurado dos adaptadores virtuales FiberChannel, uno para la red backup y otro para el acceso de peticiones al sistema.

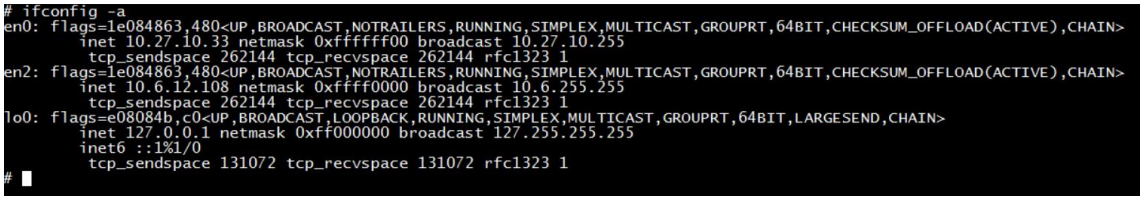

Ilustración 27 - Detalle de configuración de red (DataStage)

3.2.2 SOFTWARE

Como se apunta en la introducción de la solución de ETL, la herramienta elegida para resolver esta necesidad ha sido IBM InfoSphere DataStage, un módulo perteneciente a un producto de IBM denominado "IBM InfoSphere Information Server". Parte de la descripción del producto ya se ha abordado en el punto 2.3.3 IBM INFOSPHERE DATASTAGE.

IBM InfoSphere DataStage es una herramienta de integración de datos que permite a los usuarios mover y transformar datos entre sistemas operativos, de transacciones y analíticos.

La transformación y el movimiento de datos es el proceso mediante el cual se seleccionan, convierten y correlacionan datos de origen en el formato que requieren los sistemas de destino. El proceso manipula datos para que sean conformes con las reglas de negocio, de dominio y de integridad y con otros datos en el entorno de destino.

InfoSphere DataStage proporciona conectividad directa a aplicaciones empresariales como orígenes o destinos, garantizando que los datos más relevantes, completos y precisos se integren en el proyecto de integración de datos.

Al utilizar las funciones de proceso paralelo de plataformas de hardware multiprocesador, InfoSphere DataStage permite a la organización resolver problemas empresariales a gran escala. Se pueden procesar grandes volúmenes de datos en un proceso por lotes, en tiempo real, o como un servicio web, en función de las necesidades del proyecto.

InfoSphere DataStage está conformado por una arquitectura interna de capas. Una *capa* es un grupo lógico de componentes de InfoSphere Information Server y los sistemas en los que están instalados dichos componentes.

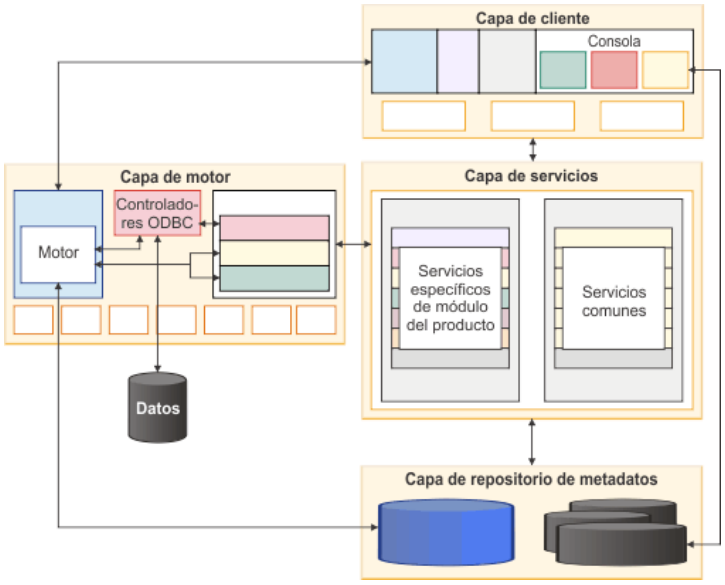

Ilustración 28 - Estructura de DataStage

3.2.2.1 Capa de Cliente

La capa de cliente se compone de los programas cliente y las consolas que se utilizan para el desarrollo, la administración y otras tareas, y los sistemas en los que están instalados.

En el caso concreto de DataStage los componentes de esta capa más utilizados son:

- Consola de IBM® InfoSphere Information Server
- Cliente de Administrador de IBM InfoSphere DataStage and QualityStage
- Cliente de IBM InfoSphere DataStage and QualityStage Designer
- Cliente de IBM InfoSphere DataStage and QualityStage Director
- Línea de mandatos istool de IBM InfoSphere Information Server. La infraestructura istool se instala en la capa de motor y la capa de cliente.
- El Gestor multicliente se instala al instalar un producto que incluya componentes de la capa de cliente de InfoSphere DataStage y InfoSphere QualityStage.

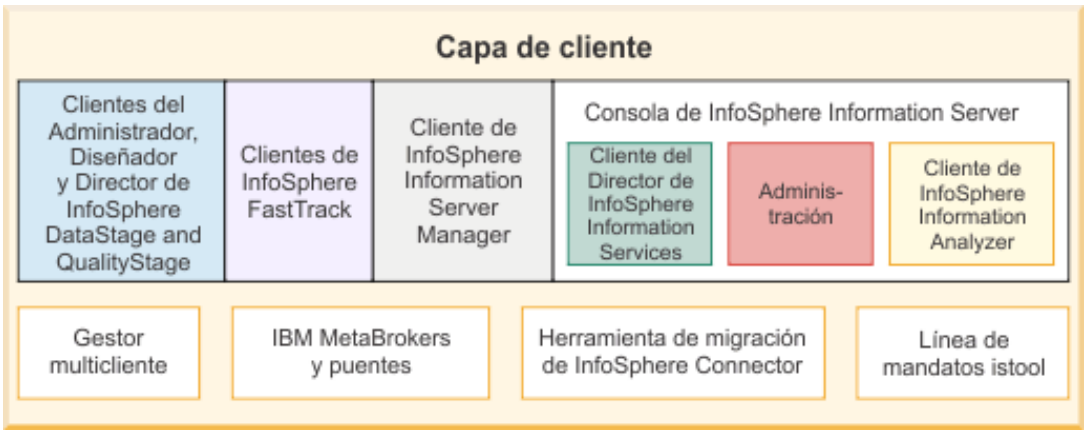

Ilustración 29 - Capa Cliente DataStage

3.2.2.2 Capa de Servicios

La capa de servicios consta del servidor de aplicaciones, los servicios comunes y los servicios de producto para la suite y los módulos de producto. Esta capa proporciona servicios comunes (por ejemplo, seguridad) y servicios que son específicos de determinados módulos de producto.

En la capa de servicios, IBM WebSphere Application Server aloja los servicios. La capa de servicios también aloja las aplicaciones de InfoSphere Information Server basadas en la Web.

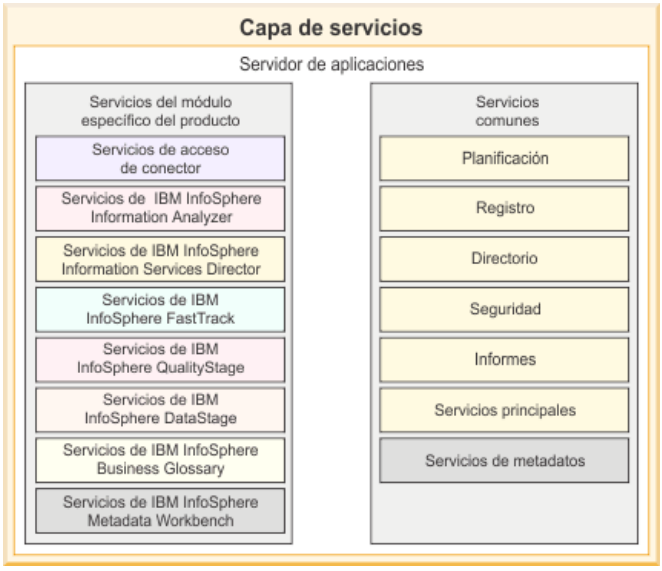

Ilustración 30 - Capa Servicios DataStage

3.2.2.3 Capa de Motor

La capa de motor es el grupo lógico de componentes de motor, agentes de comunicación, agentes de servicio, controladores ODBC, supervisión de trabajos etcétera. El motor ejecuta trabajos y otras tareas para módulos de producto.

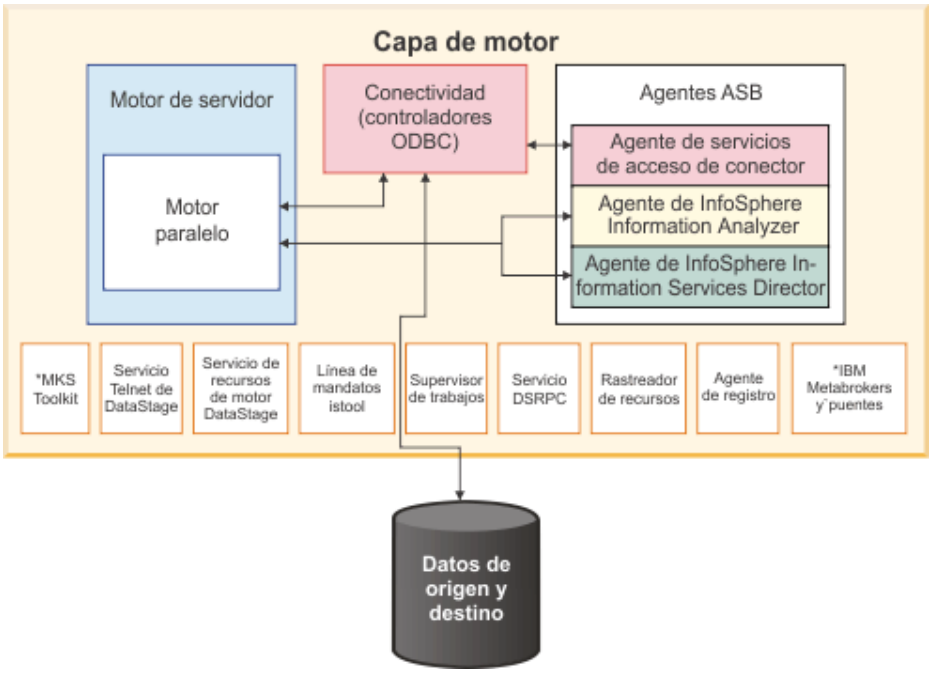

Ilustración 31 - Capa motor DastaStage

Motor de InfoSphere Information Server

Ejecuta tareas o trabajos tales como descubrimiento, análisis, limpieza o transformación. El motor incluye el motor de servidor y el motor paralelo y otros componentes de los que consta el entorno de ejecución de InfoSphere Information Server y sus componentes de producto.

Agentes ASB

Procesos Java que se ejecutan en segundo plano en cada sistema que aloja una capa de motor de InfoSphere Information Server. Cuando un servicio que se ejecuta en la capa de servicios recibe una solicitud de servicio que requiere proceso de un componente de la capa de motor, los agentes reciben y transmiten la solicitud.

AIX: Los agentes se ejecutan como daemons que se denominan ASBAgent.

Los agentes ASB incluyen:

Agente de servicios de acceso de conector

Transporta solicitudes de servicio entre los componentes de controlador ODBC de la capa de motor y el componente de servicios de acceso de conector de la capa de servicios.

Controladores ODBC

El programa de instalación instala en la capa de motor un conjunto de controladores ODBC que funciona con los componentes de InfoSphere Information Server. Estos controladores suministran conectividad a datos de origen y destino.

Rastreador de recursos

El programa de instalación instala el Rastreador de recursos para trabajos paralelos con componentes de motor para InfoSphere DataStage e InfoSphere QualityStage. El Rastreador de recursos crea registros del uso de E/S, de la memoria y del procesador en cada sistema que ejecuta trabajos paralelos.

dsrpcd (servicio DSRPC)

Permite a los clientes InfoSphere DataStage conectarse al motor de servidor.

AIX: Este proceso se ejecuta como un daemon (dsrpcd)

Supervisor de trabajos

Aplicación Java (JobMonApp) que recopila información de proceso de los trabajos del motor paralelo. La información se direcciona al proceso controlador del servidor correspondiente al trabajo de motor paralelo. El proceso controlador del servidor actualiza diversos archivos del repositorio de metadatos con estadísticas tales como el número de entradas y salidas, los recursos externos a los que se accede, la hora de inicio del operador y el número de filas procesadas.

Supervisor de metadatos operativos

Una aplicación Java (OMDMonApp) que procesa los archivos XML de metadatos operativos que generan las ejecuciones de trabajos si la recopilación de metadatos operativos está habilitada. La información de los archivos XML se almacena en el repositorio de metadatos, y los archivos XML se suprimen.

3.2.2.4 Capa Repositorio

La capa de repositorio consta del repositorio de metadatos y, si se instalan, de otros almacenes de datos que dan soporte a otros módulos de producto.

El repositorio de metadatos contiene los metadatos, datos e información de configuración de uso compartido para los módulos del producto InfoSphere Information Server. Los otros almacenes de datos almacenan datos ampliados para uso de los módulos de producto a los que dan soporte, como la base de datos de operaciones, que es un almacén de datos que utiliza la consola de operaciones del motor.

La capa de repositorio incluye la base de datos de repositorio de metadatos para InfoSphere Information Server. El repositorio de metadatos existe como su propio esquema en esta base de datos. El repositorio de metadatos es un componente compartido que almacena metadatos de tiempo de diseño, tiempo de ejecución, glosario y otros metadatos para módulos de producto de la suite InfoSphere Information Server.

Ilustración 32 - Capa de Repositorio de Metadatos

3.3 CASO DE USO: TRANSFORMACION DE PROGRAMAS RPG+COBOL EN FLUJOS DE TRABAJO DE LA ETL

Una vez que se dispone de un sistema Data Warehouse moderno y con capacidad de poder alojar los datos, y una herramienta de transformación de esos datos, hay que abordar el problema de la modernización de los programas encargados de alimentar el viejo Data Warehouse realizando su transformación a la nueva plataforma.

El sistema Data Warehouse que se va a sustituir está basado en estándares de programación con un ámbito muy específico, una complejidad elevada y que requieren de un equipo especializado lo que implica un alto coste de mantenimiento.

El sistema se basa en un almacén de datos ubicado en un servidor IBM POWER 6 con sistema operativo i5/OS. Los datos residen en una base de datos DB2.

El cliente dispone de un complejo conjunto de programas desarrollados en RPG y COBOL que se encargan de procesar los datos de entrada al sistema, realizar las transformaciones requeridas e integrar finalmente los datos en su destino.

Los datos almacenados están divididos en 3 áreas de explotación:

- **Socio Cliente**: Usuarios finales encargados del control de la fidelización de cliente, lanzamiento de ofertas, descuentos, cheques vales… así como cualquier campaña que tenga como objetivo el cliente final.
- **Gestión de Tiendas**: Usuarios encargados de llevar un control de ventas de toda la red de establecimientos del grupo, control de stock, distribución de productos, ofertas, control de personal.
- **Logística**: Usuarios finales encargados del control de mercancías, gestión de plataformas, almacenes, gestión del stock tanto en almacén como en tienda, control de los recursos empleados para el transporte de mercancías, control de pedidos de compra de mercancía a proveedores, cualquier necesidad surgida del control logístico de todo el proceso de puesta en venta de un producto.

El proceso de alimentación de las tres áreas del sistema es común y consiste en:

1. Un sistema genera un flujo de información en un determinado formato.
2. Los sistemas encargados de la gestión de comunicaciones hacen llegar esos ficheros desde los puntos origen a un sistema central donde se consolidan en ficheros de mayor tamaño.
3. La lógica desarrollada en los programas de carga de datos, se encarga de recoger los ficheros del sistema central e integrarlos en tablas temporales, ya en el sistema Data Warehouse.
4. Los procesos de transformación de datos, aplican la lógica desarrollada para alimentar las tablas finales donde reside la información de cada área, posteriormente esta información es explotada por una herramienta de Bussiness Intelligence.

El caso de uso en el que vamos a profundizar en el proyecto pertenece al área de Socio Cliente. Esta área es la que mejor representara el flujo de datos en una empresa del sector, y en ella podemos observar las mejoras sustanciales de la modernización de los sistemas.

3.3.1 Fase 1: El dato viaja de la tienda al servidor central de recepción

El flujo de datos en la fase inicial se pude describir como:

1. Cada caja de cobro de una tienda, genera un fichero con la información del detalle de cada artículo comprado agrupado por el número de ticket.
2. Una vez que la caja dispone de 5 tickets con sus detalles de cada artículo, envía un fichero al servidor central de la tienda.
3. El servidor central de la tienda agrupa los ficheros enviados por todas las cajas y según el intervalo de tiempo configurado (por ejemplo cada 5 minutos) envía un fichero con todas las ventas de la tienda, es decir, los tickets de venta con su número identificativo, así como el detalle de venta de cada artículo.
4. Todos los envíos de ficheros están gestionados por un gestor de comunicaciones que no se abarca en este proyecto.
5. El sistema central de recepción de ficheros, recibe los datos de todas las tiendas del grupo y consolida en un único fichero toda la información (este proceso no se va a mejorar en el ámbito de este proyecto)

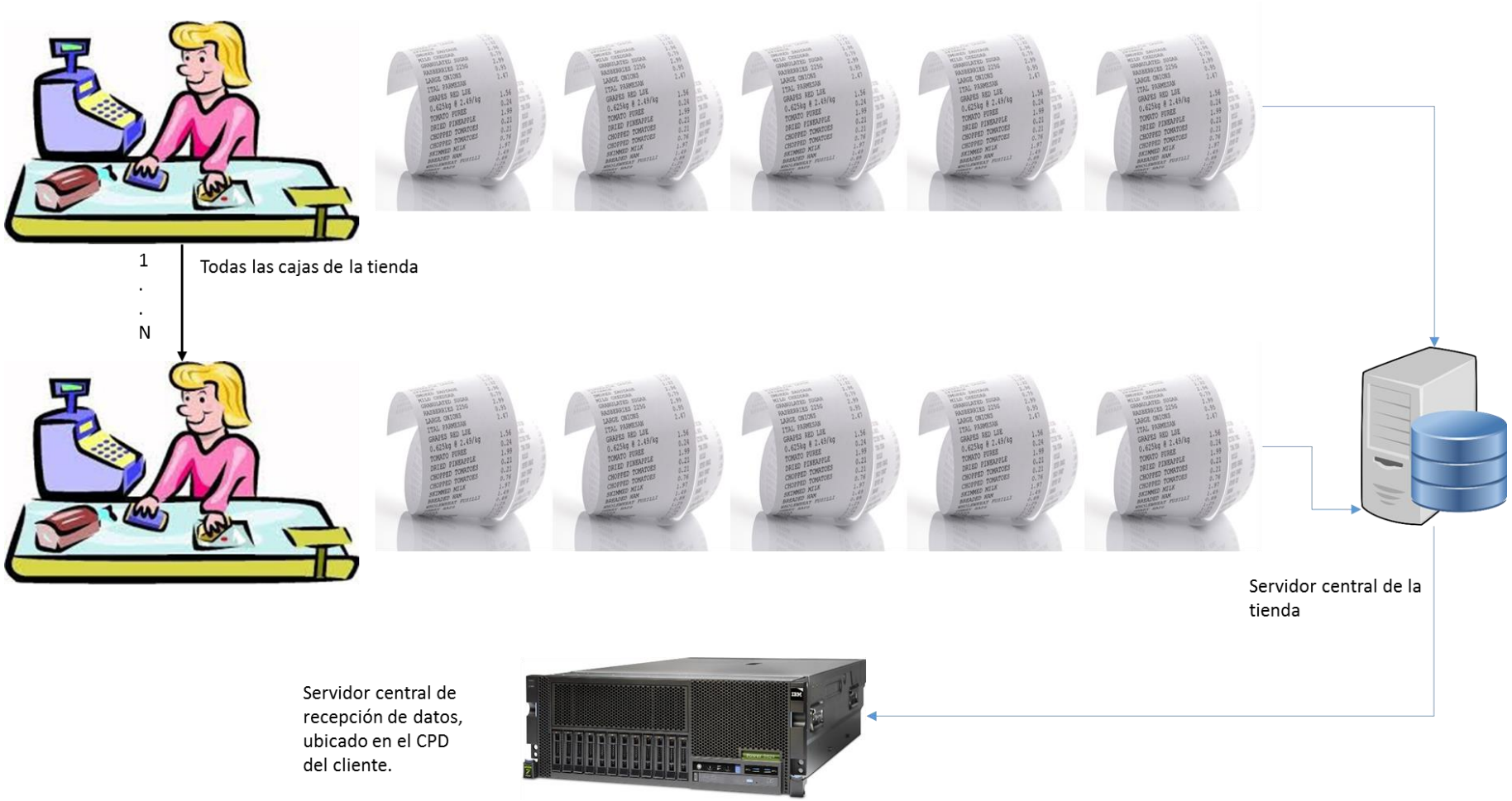

Ilustración 33 - Flujo de generación da información desde las tiendas

3.3.2 Fase 2: Integración de los ficheros de datos en tablas de Netezza

Una vez que disponemos de un fichero de datos de las tiendas ya consolidado en el servidor de recepción de ficheros, comienza el proceso de integración y transformación de los datos que se va a modernizar

En esta fase, se han transformado los programas RPG encargados de acceder a los ficheros de texto e insertar los datos en unas tablas temporales llamadas tablas de transferencia o TRF.

Para poder realizar este proceso se han tenido que llevar a cabo las siguientes tareas:

Creación de las tablas de transferencia necesarias para alojar los datos en Netezza, para ello se ha extraído el DDL de la tabla del sistema actual (DB2 sobre i5/OS) y se han realizado las transformaciones necesarias para crear la tabla en el destino.

3.3.2.1 Ejemplo de extracción del modelo de datos desde un sistema iSeries:

En primer lugar generamos un archivo DDL con las creaciones de las tablas:

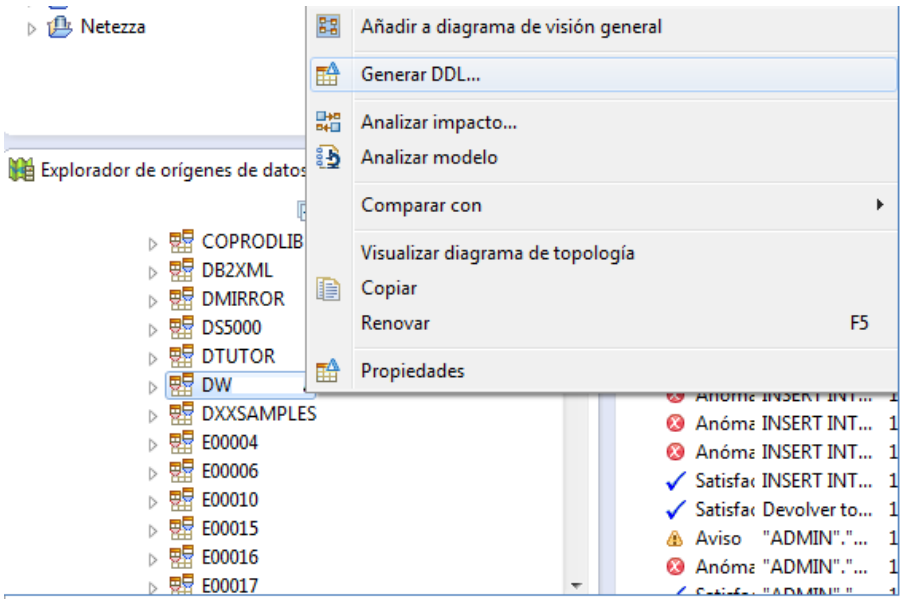

Ilustración 34 - Generación DDL iSeries (I)

Seguimos todos los pasos hasta sacar el fichero.

***IMPORTANTE:** como Netezza no soporta las sentencias LABEL del sistema iSeries, directamente no la extraemos en el DDL.

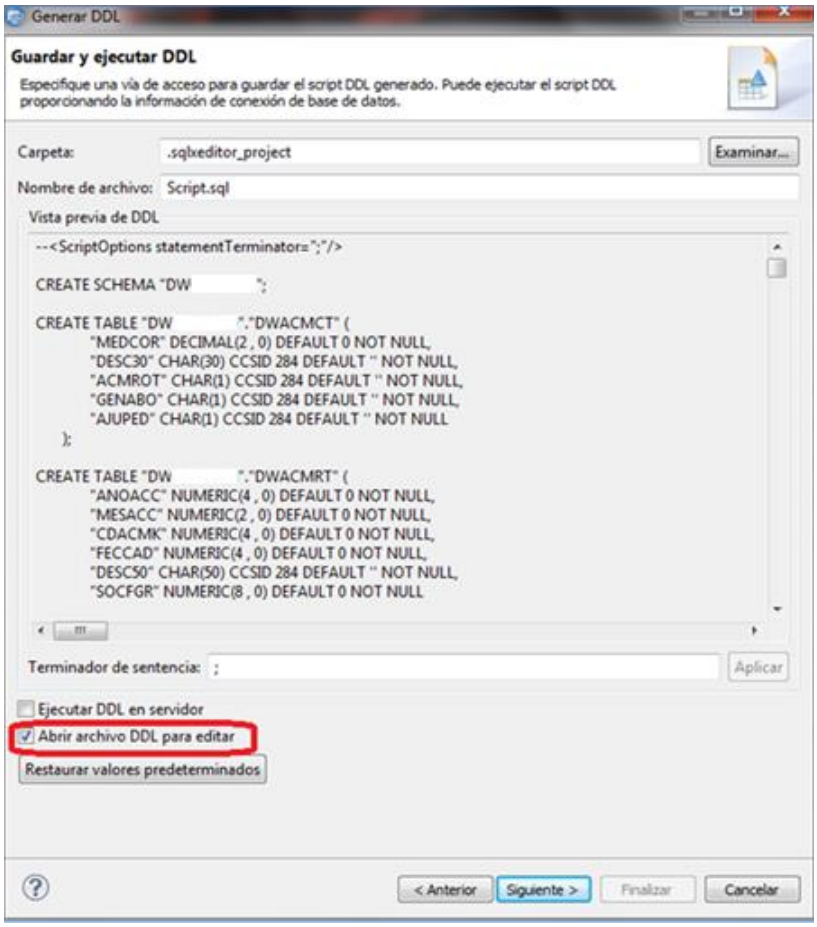

Ilustración 35 - Generación DDL iSeries (II)

A continuación copiamos el contenido de la consola y abrimos un nuevo script SQL de Netezza. El siguiente paso es copiarlo en la consola.

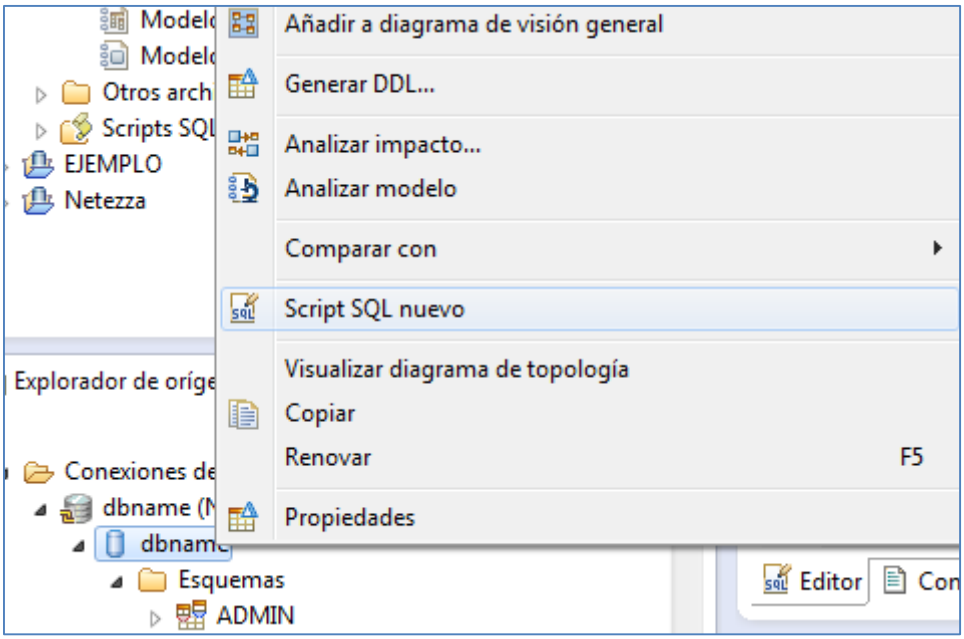

Ilustración 36 - Generación DDL iSeries (III)

A continuación se modificarán los CREATES para que Netezza los pueda interpretar:

1. CCSID 284: Netezza no soporta este CCSID del sistema iSeries por lo que si lo borramos Netezza ya no lo detecta como un error.

```
CREATE TABLE "DW      "."DWACMCT" (
        "MEDCOR" DECIMAL(2 , 0) DEFAULT 0 NOT NULL,
        "DESC30" CHAR(30) CCSID 284 DEFAULT '' NOT NULL,
        "ACMROT" CHAR(1) CCSID 284 DEFAULT '' NOT NULL,
        "GENABO" CHAR(1) CCSID 284 DEFAULT '' NOT NULL,
        "AJUPED" CHAR(1) CCSID 284 DEFAULT '' NOT NULL
    );
```

Ilustración 37 - Generación DDL iSeries (IV)

2. FOR COLUMN: el iSeries da la posibilidad de generar nombres alternativos de columnas. Es decir, dar un nombre de columna más corto. Se elimina ya que no es necesario.

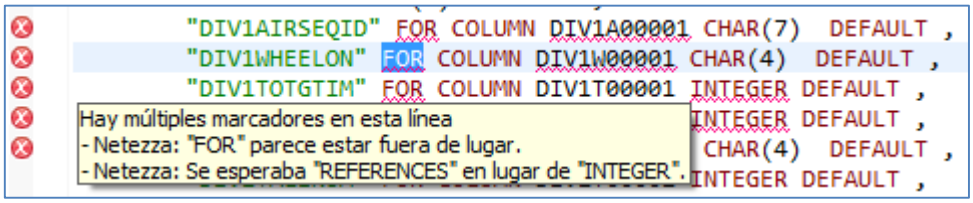

Ilustración 38 - Generación DDL iSeries (V)

Para eliminar este for colum + nombre de campo vamos a utilizar expresiones regulares:

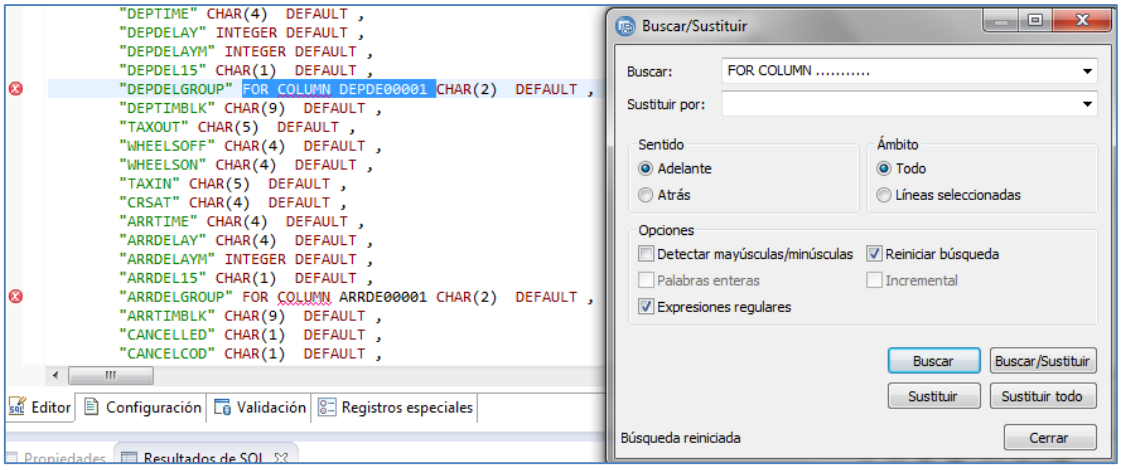

Ilustración 39 - Generación DDL iSeries (VI)

3. LABEL ON: en Netezza no existen las etiquetas “Label on”. Por este motivo, es necesario reemplazarlas por “COMMENT ON”.

4. Optimización de Numeric y Decimal: vamos a modificar este tipo de datos para optimizar los Zone Maps de Netezza. Para ello, transformaremos todos los Numeric y Decimal a Integers en función de su tamaño:

   - Decimal (1,0) / Numeric (1,0): byteint
   - Decimal (2,0) / Numeric (2,0): byteint
   - Decimal (3,0) / Numeric (3,0): smallint
   - Decimal (4,0) / Numeric (4,0): smallint
   - Decimal (5,0) / Numeric (5,0): integer
   - …
   - Decimal (10,0) / Numeric (10,0): bigint
   - …

5. Distribución de las tablas: por defecto vamos a poner una distribución random:

) DISTRIBUTE ON RANDOM;

6. Sustituir TEXT IS por IS.

3.3.2.2 Ejemplo de creación de modelo de datos en Netezza

En el siguiente código se muestra un ejemplo de la creación de las tablas en la base de datos de Netezza, ejecutando el script generado en la fase previa:

```
CREATE TABLE "SOCIOCDB"."DWBPTPT" (
        "OPPIDT" INTEGER NOT NULL,
        "OPPAFI" SMALLINT NOT NULL,
        "OPPMFI" BYTEINT NOT NULL,
        "OPPNTA" CHAR(22) NOT NULL,
        "OPPIMP" DECIMAL(11 , 3) NOT NULL,
        "OPPNPA" SMALLINT NOT NULL,
        "OPPFCR" INTEGER NOT NULL,
        "OPPFMR" INTEGER NOT NULL
    ) DISTRIBUTE ON (OPPIDT);
```

Ilustración 40 - Generación Tablas Netezza (I)

En el siguiente código se muestra un ejemplo de la creación de los comentarios, de tablas en la base de datos de Netezza:

```
COMMENT ON TABLE "SOCIOCDB"."DWACMCT" IS
'DW.Medidas Correctoras.Physical table';

COMMENT ON COLUMN "SOCIOCDB"."DWACMCT"."MEDCOR" IS
'Medida Correctora';

COMMENT ON COLUMN "SOCIOCDB"."DWACMCT"."MEDCOR" IS
'Medida Correctora';

COMMENT ON COLUMN "SOCIOCDB"."DWACMCT"."DESC30" IS
'Descripcion Generica';

COMMENT ON COLUMN "SOCIOCDB"."DWACMCT"."DESC30" IS
'Descripcion Generica';

COMMENT ON COLUMN "SOCIOCDB"."DWACMCT"."ACMROT" IS
'Acumular Roturas';

COMMENT ON COLUMN "SOCIOCDB"."DWACMCT"."ACMROT" IS
'Acumular Roturas';
```

Ilustración 41 - Generación Tablas Netezza (II)

Una vez que disponemos de las tablas necesarias para poder alojar los datos provenientes de los ficheros de texto es necesario generar un flujo de trabajo que se encargue de integrarlos.

3.3.2.3 Carga de tablas temporales (CARGATRF)

Esquema de la secuencia desarrollada en la ETL, encargada de alimentar las tablas de transferencia

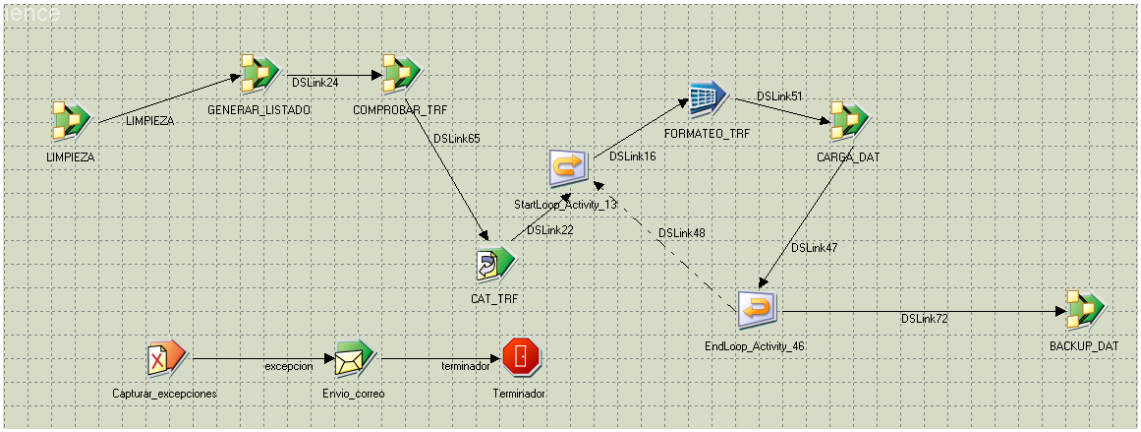

Ilustración 42 - DataStage Job: Carga de tablas temporales (TRF)

Como se puede observar la secuencia está compuesta por varias etapas de distintos tipos

La secuencia LIMPIEZA se encarga de truncar las tablas de transferencia antes de realizar una nueva carga de datos, así como de limpiar los ficheros temporales que utiliza el proceso.

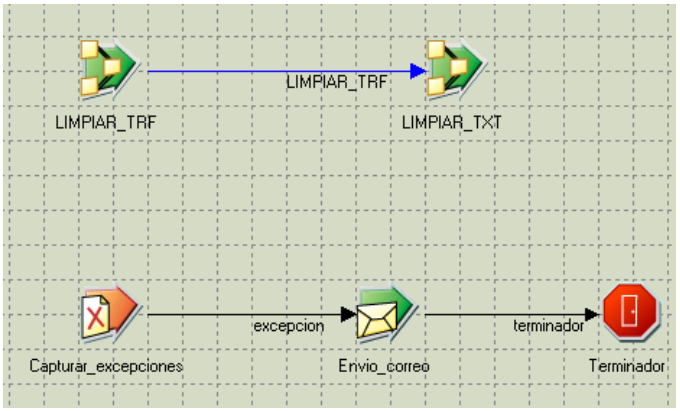

Ilustración 43 - DataStage Job: LIMPIEZA (CARGATRF)

LIMPIAR_TRF: Truncate de las tablas temporales, realizando una comprobación de la correcta finalización del proceso. Utilizando scripts de Shell y trabajos de lectura de base de datos.

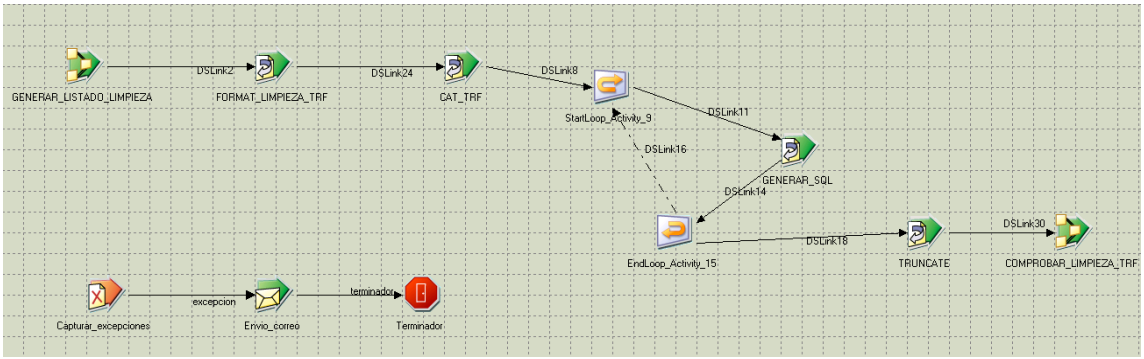

Ilustración 44 - DataStage Job: LIMPIAR_TRF (LIMPIEZA)

LIMPIAR_TXT: Proceso que limpiar los ficheros temporales donde se guarda un log de los ficheros que se integran en cada proceso de carga de trf

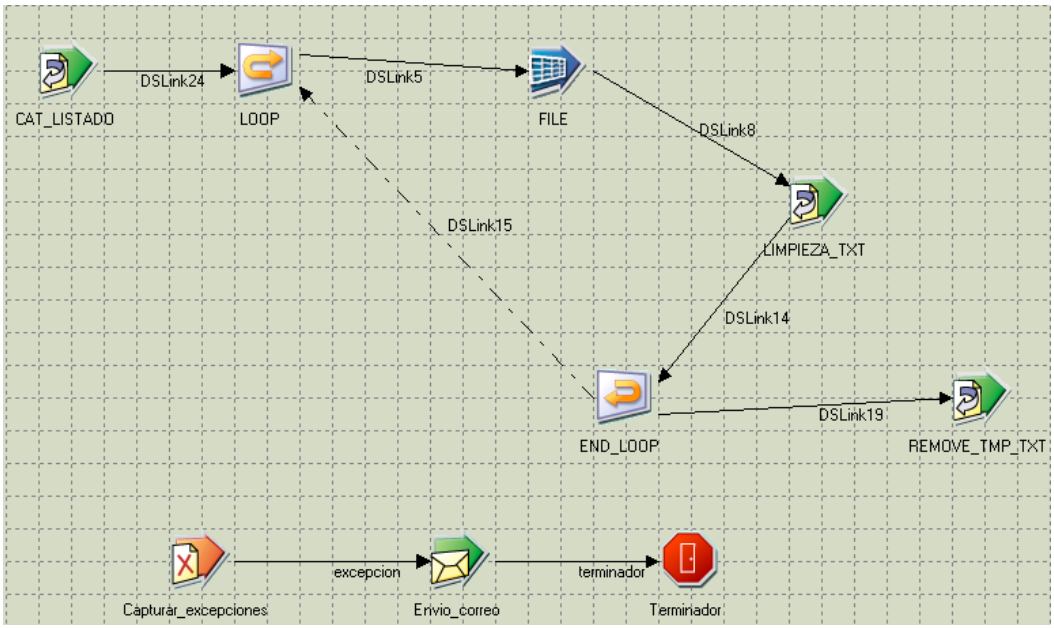

Ilustración 45 - DataStage Job: LIMPIAR_TXT (LIMPIEZA)

Una vez que tenemos las tablas vacías y los ficheros eliminados comenzamos el proceso de integración de los datos, para ello lo primero es obtener un listado de los ficheros que tenemos que ir a buscar al sistema central de recepción de datos.

El trabajo GENERAR_LISTADO es el encargado de realizar una consulta a la tabla donde está la información de los prefijos de los ficheros que hay que buscar y volcar esos datos a un fichero de texto.

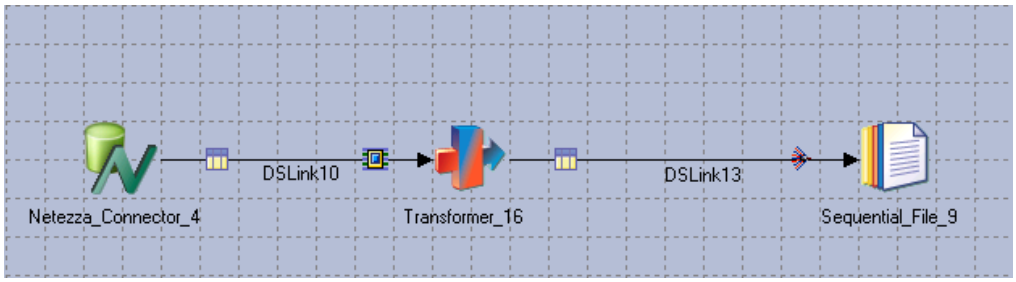

Ilustración 46 - DataStage Job: GENERAR_LISTADO (CARGATRF)

A partir del listado generado se lanza una secuencia COMPROBAR_TRF, que se encarga de conectarse al sistema central de recepción de datos, mediante conexiones FTP y ajustar el listado inicial a un listado definitivo con los prefijos de los que existen ficheros para integrar.

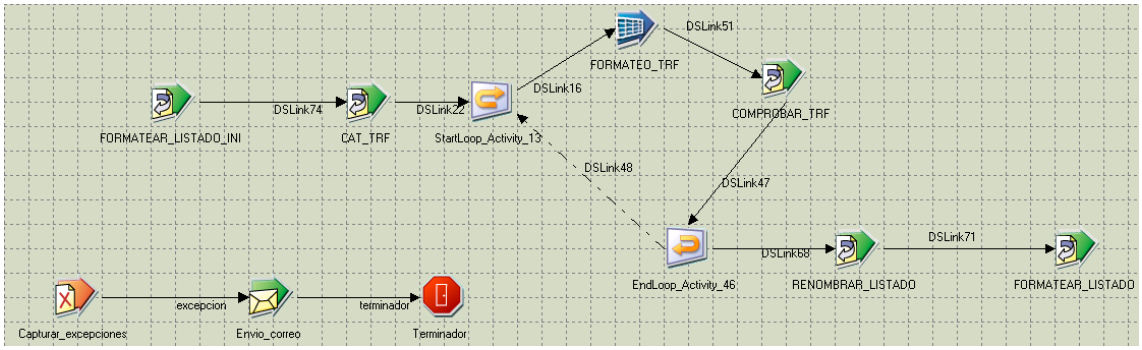

Ilustración 47 - DataStage Job: COMPROBAR_TRF (CARGATRF)

Como resultado de salida de esta comprobación tenemos un fichero con los prefijos de los que tenemos ficheros disponibles para integrar.

La secuencia de CARGATRF continúa y ejecuta un bucle con el listado de prefijos disponibles como iteraciones, de esta forma en cada ejecución del trabajo contenido en el bucle se va a proceder a buscar un tipo o prefijo diferente de ficheros.

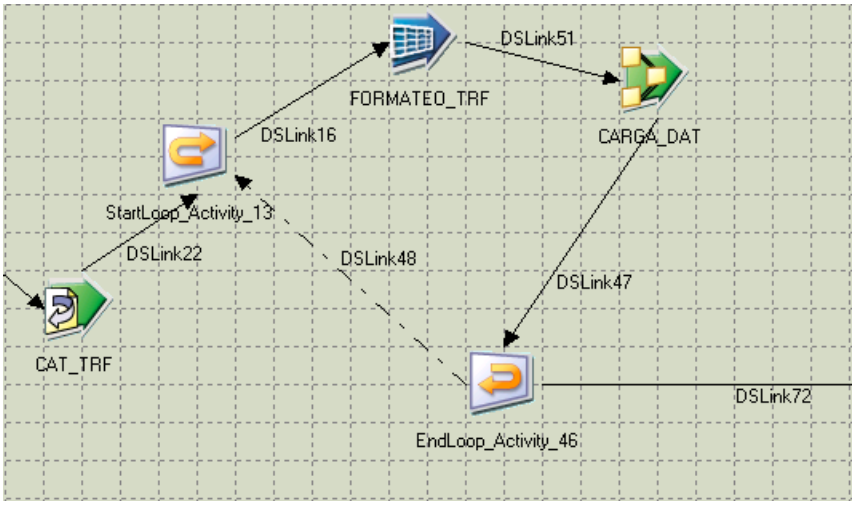

Ilustración 48 - DataStage Job: BUCLE (CARGATRF)

El trabajo CARGA_DAT es el encargado de llamar a cada uno de los trabajos individuales asociados a cada prefijo que son los encargados de realizar la conexión FTP contra el sistema central y directamente sin traer el fichero al servidor local datastage, de integrar los datos en la tabla TRF correspondiente.

En la siguiente ilustración se muestra, un esquema, no completo, del job, ya que al tener que comprobar 32 posibilidades distintas de nombres de ficheros el tamaño del trabajo es bastante extenso.

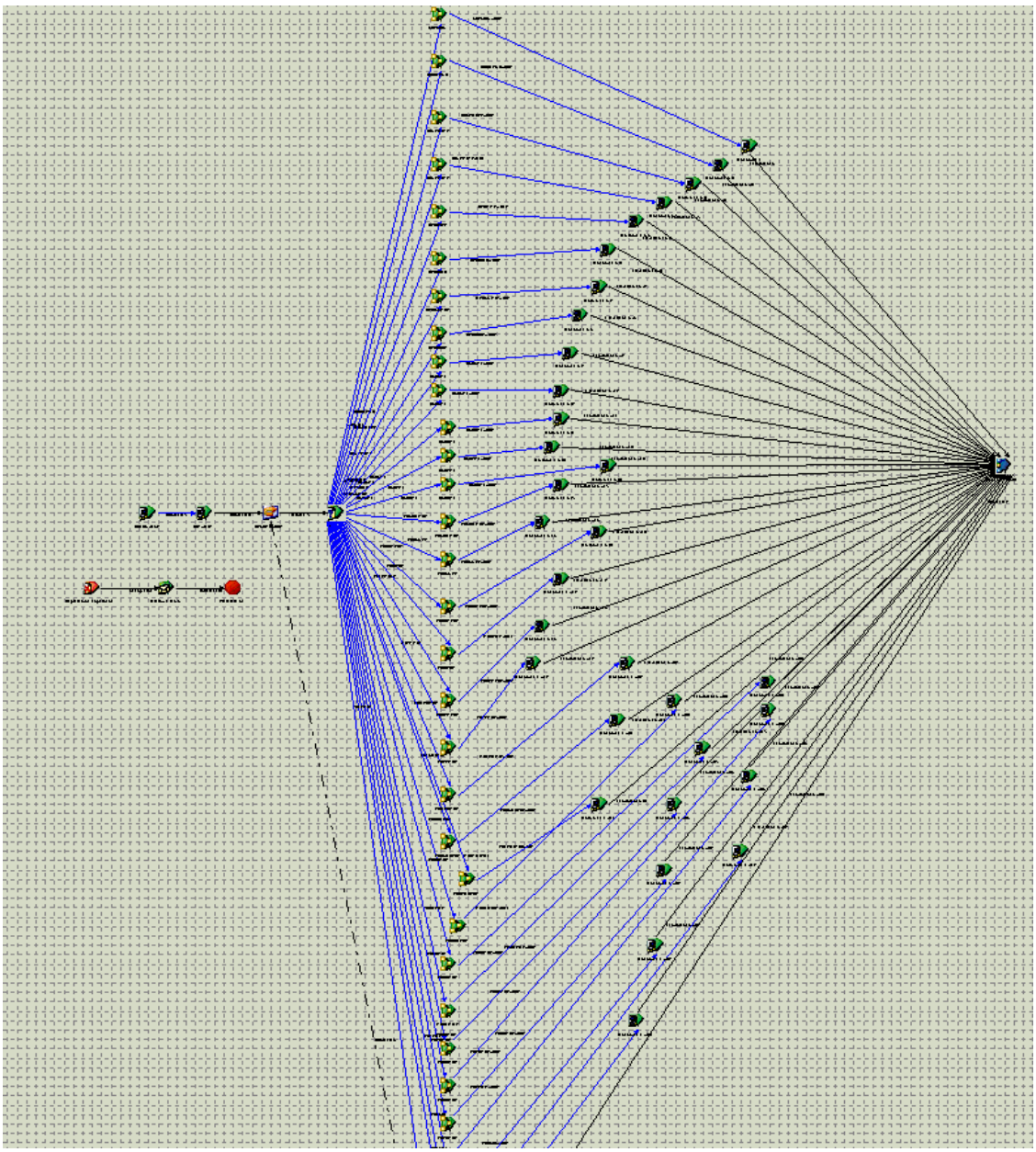

Ilustración 49 - DataStage Jobs: CARGA_DAT (CARGATRF)

En cada ejecución, el trabajo lee el prefijo y la ubicación de los ficheros asociados a él y mediante una etapa de datastage semeja a un case, comprueba que tipo de prefijo es y ejecuta el trabajo asociado a él.

Como ya se ha comentado, en el área de socio cliente, existen 32 prefijos, o tipos diferentes de ficheros que contienen los datos necesarios para alimentar al Data Warehouse

Listado de trabajos encargados de la integración de los ficheros a las tablas de transferencia:

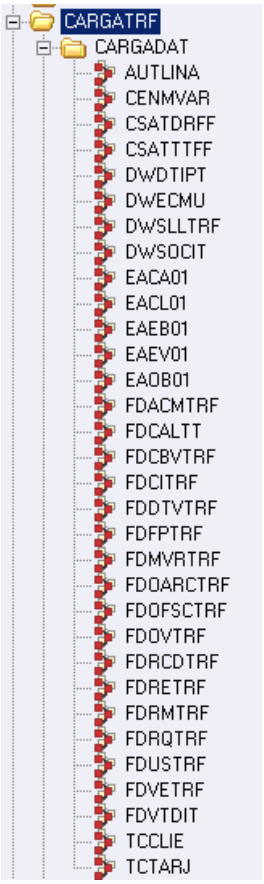

Ilustración 50 - DataStage Job: Listado Trabajos CARGADAT

Unos de los prefijos del sistema es CSATDRFF que alimenta una tabla de transferencia que utilizaremos en la siguiente fase.

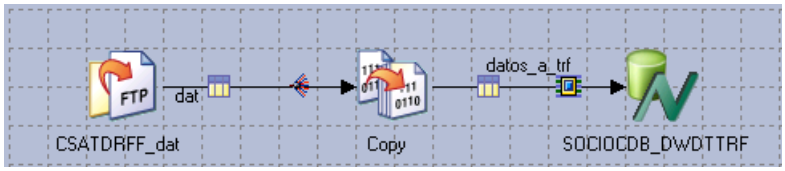

Ilustración 51 - DataStage Job: CSATDRFF (CARGADAT)

Este trabajo utiliza una etapa FTP de DataStage para acceder al contenido del fichero en el sistema remoto y extraer su contenido, ajustar tipos de datos con la etapa Copy y finalmente escribir los registros en la tabla trf destino asociado al prefijo CSATDRFF en este caso DWDTTRF.

Otro ejemplo de trabajo de carga de trf es el asociado al prefijo AUTLINA

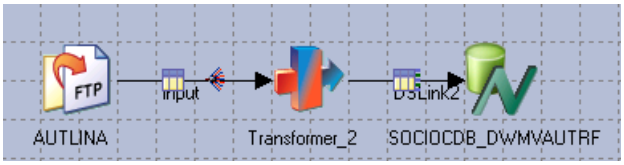

Ilustración 52 - DataStage Job: AUTLINA (CARGADAT)

En este caso los datos requieren de una transformación por lo que es necesario incluir una etapa de TRANSFORMER de DataStage para poder realizar las transformaciones requeridas.

Si accedemos a la etapa de transformer podemos ver el detalle del flujo de los datos y las transformaciones aplicadas a ellos.

En este caso, aplicamos una función de cambio de tipo de datos StringToDecimal para concatenar dos substring de un campo con un carácter.

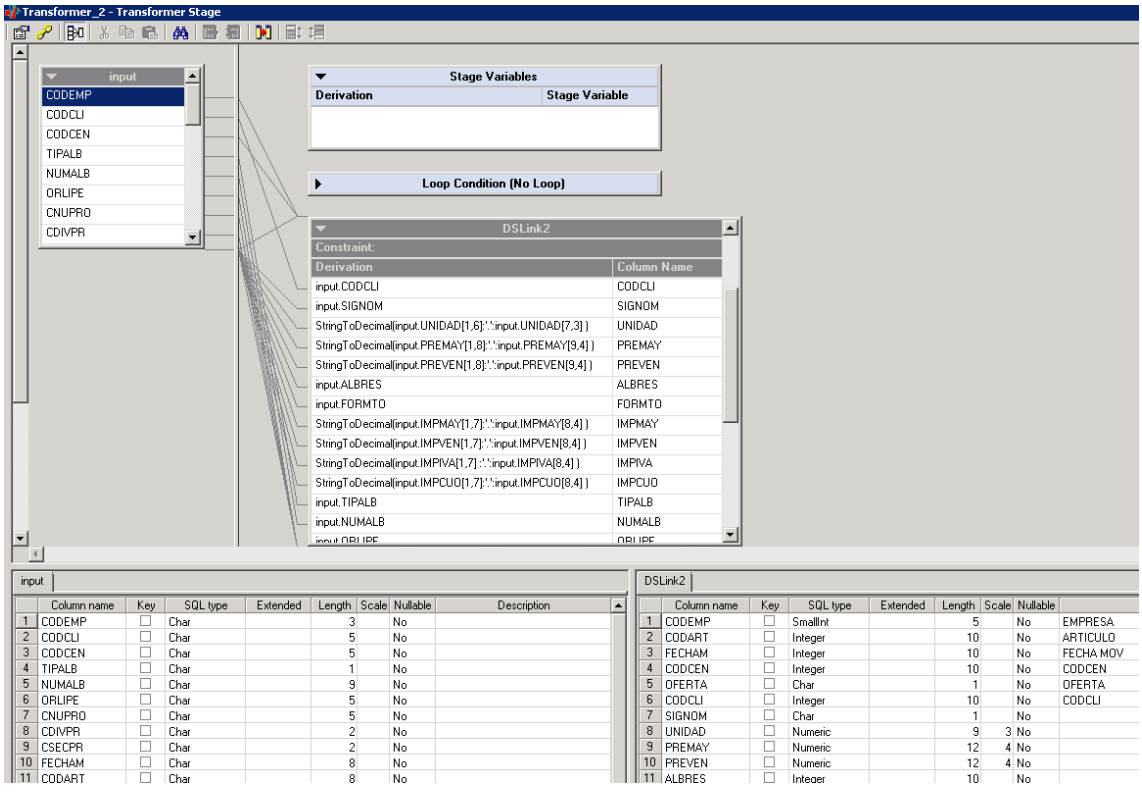

Ilustración 53 - DataStage Job: Detalle Transformer (AUTLINA)

El proceso de CARGADAT, también se encarga de, remotamente, mover los ficheros ubicados en el servidor central, del directorio origen a un directorio de procesados en el que permanecen durante 1 mes por si es necesario restaurar una tabla y volver a cargar los datos.

Como punto final a la fase de integración de datos, se realiza un backup de los ficheros de los prefijos en los que se dispone de la información de cada uno de los ficheros procesados.

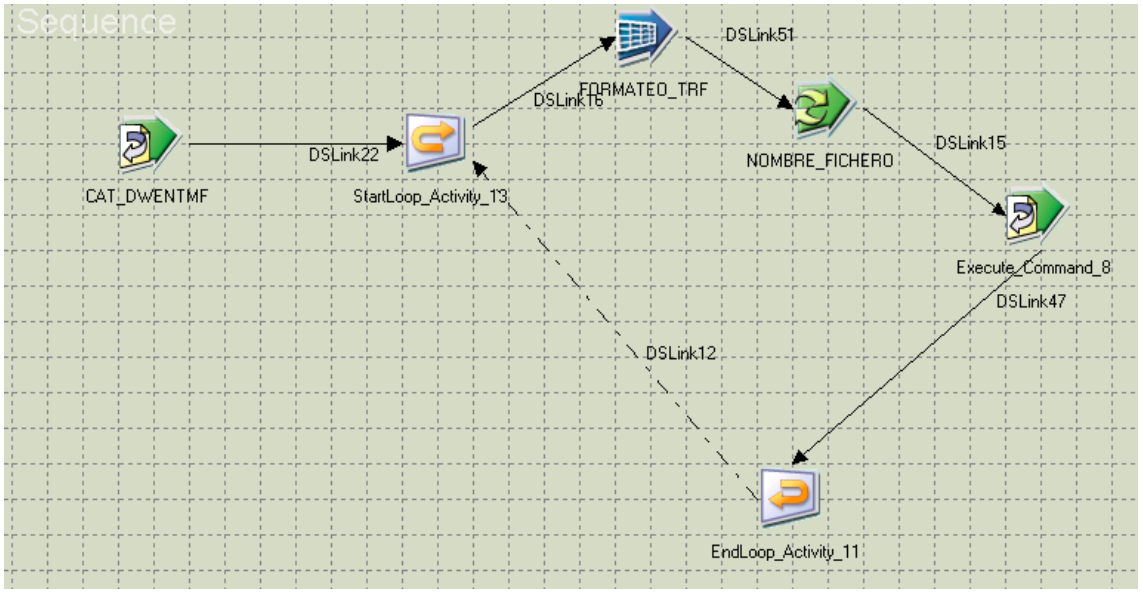

Ilustración 54 - DataStage Job: BACKUPDAT (CARGATRF)

Este proceso genera una carpeta en un directorio local y graba un fichero por prefijo con la fecha y hora de integración de los datos.

Ejemplo de fichero backup para el prefijo CSATDRFF

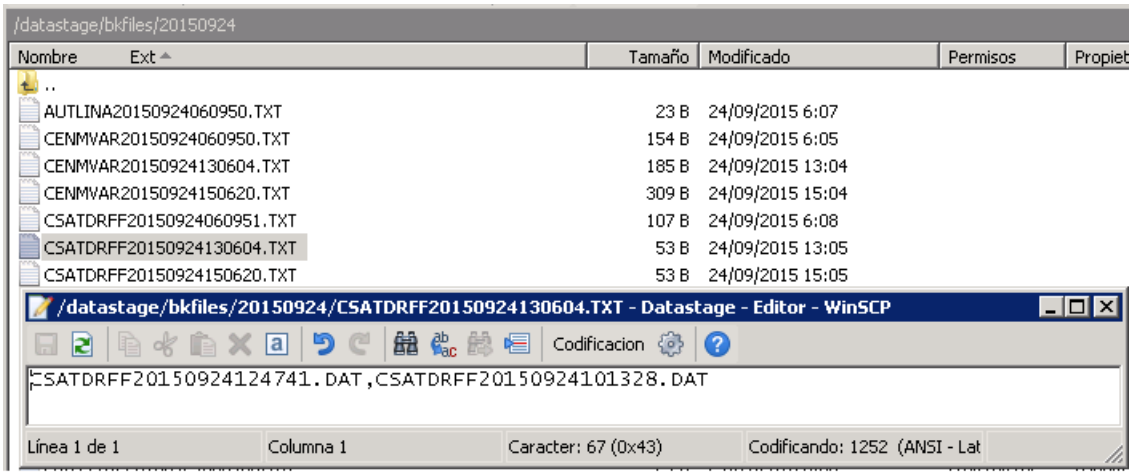

Ilustración 55 - DataStage Job: BACKUPDAT Ejemplo fichero (CARGATRF)

3.3.3 Fase 3: Transformación de los datos aplicando la lógica de negocio

Una vez que disponemos de los datos que queremos integrar en las tablas de transferencia, tenemos que analizar la lógica desarrollada en los programas RPG y COBOL y transformarla en un trabajo que la ETL pueda ejecutar correctamente.

Vamos a ver el proceso completo para uno de los programas del área de socio cliente, este programa se encarga de integrar los datos de los movimientos de los tickets de todas las tiendas del grupo, es un resumen de la información del ticket.

Lo primero que necesitamos realizar es un análisis del código fuente del programa y extraer un pseudocodigo que podamos utilizar para el desarrollo con la herramienta ETL.

Del análisis tenemos que extraer los siguientes puntos clave:

- Nombre y ubicación de la tabla de transferencia origen, nombre de las columnas y tipos de datos asociados
- Nombre y ubicación de la tabla de datos consolidados final, nombre de las columnas y tipos de datos asociados
- Correspondencia de columnas de datos de origen con columnas de datos destino, lo normal es que no tengan el mismo nombre.
- Identificación de las columnas clave destino, para poder realizar acciones como update, delete o comprobación de registros duplicados.
- Cualquier campo, calculado mediante join con otras tablas, explicación de cómo extraer el campo, porque claves igualar.
- Todas las transformaciones que se tengan que realizar sobre los datos, operaciones aritméticas, cambio de tipos de datos… etc.

3.3.3.1 Análisis del programa

Tras el análisis del código del programa de estudio obtenemos los siguientes datos con los que tenemos que diseñar el trabajo que satisfaga toda la lógica.

En la siguiente tabla vemos el resumen de los campos de origen, con sus tipos de datos.

SOCIOCDB.DWTKTRF - MVTOS TOTALES TICKET TXT

Campo	Tipo	Descripción
CENTR2_TXT	CHAR(4)	Numero de Centro 2 txt
FECHA_TXT	CHAR(6)	Fecha Movimiento txt
HORA_TXT	CHAR(4)	Hora Movimiento txt
NUMTIC_TXT	CHAR(6)	Numero de Ticket txt
TIFACT	CHAR(1)	Tipo de Factura
SIGNO_TIC	CHAR(1)	Signo ticket
IMPORTE1_T	CHAR(11)	Importe 1 txt
TIP_TARJ	CHAR(1)	Tipo tarjeta
SIGNO_IMTA	CHAR(1)	Signo impte tarjeta
IMPTARJ_T	CHAR(11)	Importe tarjeta txt
NUMTAR22	CHAR(22)	Número de Tarjeta 22
SIGNO_DTO	CHAR(1)	Signo dto
IMPDESC_T	CHAR(11)	Importe descuento txt
DESCU_TXT	CHAR(4)	Porcentaje descuento txt
TARJSCTXT	CHAR(13)	Tarjeta fidelizacion txt
CAJA_TXT	CHAR(3)	Caja txt
CAJERA_TXT	CHAR(4)	Cajera txt
SINMON	CHAR(1)	Moneda
SIGNO_CUO	CHAR(1)	Signo cuota
IMPCUO_T	CHAR(11)	Importe cuota txt
SIGNO_CUP	CHAR(1)	Signo cupon
IMPCUP_T	CHAR(11)	Importe cupon txt
SIGNO_INV	CHAR(1)	Signo inversion
IMPINV_T	CHAR(11)	Importe inversion txt
DW3ZA	CHAR(8)	Campo Relleno 8

Tabla 5 - Datos origen (Movimientos Ticket)

Obtenemos también los datos asociados a la tabla de destino final.

SOCIOCDB.DWTTTKT - MVTOS TOTALES TICKET

Campo	Tipo	Descripción
CODEMP	DECIMAL(3, 0)	Código de Empresa (K)
CODCLI	DECIMAL(5, 0)	Cliente (K)
CODCEN	DECIMAL(4, 0)	Centro (K)
FECHA	DECIMAL(8, 0)	Fecha Movimiento (K)
HORA	NUMERIC(4, 0)	Hora Movimiento (K)
NUMTIC	NUMERIC(6, 0)	Numero de Ticket (K)
TIFACT	CHAR(1)	Tipo de Factura (K)
IMPORTE1	NUMERIC(11, 3)	Importe 1
TIP_TARJ	CHAR(1)	Tipo tarjeta
IMPTARJ	NUMERIC(11, 3)	Importe tarjeta
NUMTAR22	CHAR(22)	Número de Tarjeta 22
IMPDESC	NUMERIC(11, 3)	Importe descuento
PORDESC	NUMERIC(5, 2)	Porcentaje descuento
TARJSCTXT	CHAR(13)	Tarjeta fidelización
CAJA	NUMERIC(3, 0)	Caja
CAJERA	NUMERIC(4, 0)	Cajera
IMPCUO	NUMERIC(11, 3)	Importe cuota
IMPCUP	NUMERIC(11, 3)	Importe cupón
IMPINV	NUMERIC(11, 3)	Importe inversión
FACTU1	DECIMAL(8, 0)	Fecha Act1
IDTIEN	DECIMAL(7, 0)	Id interno de la tienda
IDSOCI	DECIMAL(7, 0)	Id interno del socio
HORFIS	DECIMAL(2, 0)	Hora de 0 a 23
DIAFIS	DECIMAL(2, 0)	Dia del mes
SEMFIS	DECIMAL(2, 0)	Semana fiscal
MESFIS	DECIMAL(2, 0)	Mes fiscal
ANOFIS	DECIMAL(4, 0)	Año fiscal
FECPRO	DECIMAL(8, 0)	Fecha proceso
DETPRO	CHAR(30)	Detalle proceso

Tabla 6 - Datos final (Movimientos Ticket)

Las transformaciones que tenemos que realizar son las siguientes:

SOCIOCDB.DWTKTRF - MVTOS TOTALES TICKET TXT

Campo	Tipo	Descripción
CENTR2_TXT	CHAR(4)	Numero de Centro 2 txt
FECHA_TXT	CHAR(6)	Fecha Movimiento txt
HORA_TXT	CHAR(4)	Hora Movimiento txt
NUMTIC_TXT	CHAR(6)	Numero de Ticket txt
TIFACT	CHAR(1)	Tipo de Factura
SIGNO_TIC	CHAR(1)	Signo ticket
IMPORTE1_T	CHAR(11)	Importe 1 txt
TIP_TARJ	CHAR(1)	Tipo tarjeta
SIGNO_IMTA	CHAR(1)	Signo impte tarjeta
IMPTARJ_T	CHAR(11)	Importe tarjeta txt
NUMTAR22	CHAR(22)	Numero de Tarjeta 22
SIGNO_DTO	CHAR(1)	Signo dto
IMPDESC_T	CHAR(11)	Importe descuento txt
DESCU_TXT	CHAR(4)	Porcentaje descuento txt
TARJSCTXT	CHAR(13)	Tarjeta fidelizacion txt
CAJA_TXT	CHAR(3)	Caja txt

SOCIOCDB.DWTTTKT - MVTOS TOTALES TICKET

Campo	Tipo	Descripción	Operación
CODEMP	DECIMAL(3, 0)	Codigo de Empresa (K)	1
CODCLI	DECIMAL(5, 0)	Cliente (K)	**(*3)**
CODCEN	DECIMAL(4, 0)	Centro (K)	**(*3)**
FECHA	DECIMAL(8, 0)	Fecha Movimiento (K)	= Añadir siglo (AAAAMMDD)
HORA	NUMERIC(4, 0)	Hora Movimiento (K)	=
NUMTIC	NUMERIC(6, 0)	Numero de Ticket (K)	=
TIFACT	CHAR(1)	Tipo de Factura (K)	=
IMPORTE1	NUMERIC(11, 3)	Importe 1	(IMPORTE1_T * SIGNO_TIC)/1000
TIP_TARJ	CHAR(1)	Tipo tarjeta	=
IMPTARJ	NUMERIC(11, 3)	Importe tarjeta	(IMPTARJ_T *SIGNO_IMTA)/1000
NUMTAR22	CHAR(22)	Numero de Tarjeta 22	=
IMPDESC	NUMERIC(11, 3)	Importe descuento	(IMPDESC_T * SIGNO_DTO)/1000
PORDESC	NUMERIC(5, 2)	Porcentaje descuento	=
TARJSCTXT	CHAR(13)	Tarjeta fidelizacion	= (Si TARJSCTXT = '0000000000000' poner campo a blancos)
CAJA	NUMERIC(3, 0)	Caja	=

CAJERA_TXT	CHAR(4)	Cajera txt		CAJERA	NUMERIC(4, 0)	Cajera	=
SINMON	CHAR(1)	Moneda					
SIGNO_CUO	CHAR(1)	Signo cuota					
IMPCUO_T	CHAR(11)	Importe cuota txt		IMPCUO	NUMERIC(11, 3)	Importe cuota	(IMPCUO_T *SIGNO_CUO)/1000
SIGNO_CUP	CHAR(1)	Signo cupon					
IMPCUP_T	CHAR(11)	Importe cupon txt		IMPCUP	NUMERIC(11, 3)	Importe cupon	(IMPCUP_T * SIGNO_CUP)/1000
SIGNO_INV	CHAR(1)	Signo inversion					
IMPINV_T	CHAR(11)	Importe inversion txt		IMPINV	NUMERIC(11, 3)	Importe inversion	(IMPINV_T * SIGNO_INV)/1000
DW3ZA	CHAR(8)	Campo Relleno 8					
				FACTU1	DECIMAL(8, 0)	Fecha Act1	0
				IDTIEN	DECIMAL(7, 0)	Id interno de la tienda	(*4)
				IDSOCI	DECIMAL(7, 0)	Id interno del socio	**(*2)**
				HORFIS	DECIMAL(2, 0)	Hora de 0 a 23	Dos primeros caracteres de HORA_TXT
				DIAFIS	DECIMAL(2, 0)	Dia del mes	DIAFIS **(*1)**
				SEMFIS	DECIMAL(2, 0)	Semana fiscal	SEMFIS **(*1)**
				MESFIS	DECIMAL(2, 0)	Mes fiscal	MESFIS **(*1)**
				ANOFIS	DECIMAL(4, 0)	Año fiscal	ANOFIS **(*1)**
				FECPRO	DECIMAL(8, 0)	Fecha proceso	Fecha actual
				DETPRO	CHAR(30)	Detalle proceso	' '

Tabla 7 - Transformaciones (Movimientos Ticket)

Como se puede ver en la tabla de transformaciones, no todas son operaciones, literales, o cambios de datos. Hay operaciones de búsqueda de campos en tablas foráneas. Estas transformaciones son las siguientes:

(*1) Acceder a la tabla NETEZZA.SOCIOCDB.CALENDAR con FECHA = FECHA_TXT y recuperar ANOFIS, MESFIS, SEMFIS y DIAFIS.

(*2)
IDSOCI = 0
Si TARJSCTXT <> ' ' --> **Tarjeta Fidelización**
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTFI = TARJSCTXT
Si IDSOCI = 0 --> **Tarjeta Fidelización Provisional**
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTFP = TARJSCTXT
Si IDSOCI = 0 --> **NIF**
TPJ = Primer carácter de TARJSCTXT si es una letra
IFI = letra del NIF
CPF = número NIF (9 caracteres a partir de la 1ª ó 2ª posición de TARJSCTXT)
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTPJ = TPJ AND
SOCIFI = IFI AND
SOCCPF = PF
(Si aunque hayamos encontrado el socio, éste no tiene tarjeta fidelización,
entonces ponemos id genérico)
Si dwsocit.soctfi =' ' AND dwsocit.soctpf=' '
entonces: IDSOCI = 9999999999
Si IDSOCI = 0 --> **ID Genérico**
IDSOCI = 9999999999
Sino
Si NUMT22 <> ' ' --> **Tarjeta Pago**
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTPA = NUMT22
TARJSCTXT= SOCTFI
FinSi
FinSi

(*3)
Acceder a la tabla **GENTDATF** con CODCNT=CENTR2_TXT
y obtener CODCLI y CODCEN.

(*4)
Si CODCLI>0
Acceder a DWTIENT con TIECEN = CENTR2_TXT y recuperar IDTIEN
sino
Acceder a DWTIENT con TIECEN = CODCEN y recuperar IDTIEN
finsi;

Tenemos también otros dos supuestos que hay que tener en cuenta:

1. Solo podemos procesar los tickets que no existan en un control de ticket que se almacena en la tabla DWCTTKT.

   El análisis detallado seria el siguiente:

   Para todos los registros de entrada acceder a tabla DWCTTKT y para todos los registros con TIPPROC='ACT0004' comprobar que no existan coincidencias por la siguiente clave, CODEMP, CODCLI, CODCEN, FECHA, HORA, CAJA, NUMTIC

   Al finalizar el trabajo adicionalmente a la escritura de los datos en la tabla final de totales, hay que insertar todos los registros procesados en la tabla del control de tickets. De esta forma si por algún error en la carga de datos volvemos a procesar información ya integrada, esta será desechada no corrompiendo los datos del sistema.

2. Los datos relacionados con los tickets también alimentan una serie de agregados o sumarizados que se encuentran en el sistema, debido a esto tenemos que tener en cuenta que hay que copiar los datos de los tickets integrado para que el proceso encargado de realizar los sumarizados pueda funcionar.

Una vez que tenemos definido el origen de datos, las transformaciones que tenemos que realizar, el formato de los datos finales resultantes y todos los supuestos adicionales de integración con el resto de trabajos, tenemos que desarrollar el trabajo o trabajos necesarios utilizando la herramienta ETL, InfoSphere DataStage.

Para poder realizar esta tarea utilizaremos el software de escritorio InfoSphere DataStage Designer.

3.3.3.2 Diseño de la solución

Después de tener en cuenta el análisis del programa y de los supuestos adicionales, vemos que es necesario crear dos trabajos, en una secuencia, para poder realizar todas las tareas necesarias.

Esto es debido a que el supuesto relacionado con la existencia o no de los tickets de origen en la tabla de control de tickets, nos obliga realizarlo en SQL, para que sea eficiente, ya que la tabla de control tiene un número de registros elevado >350.000.000 y un crecimiento medio >100.000 registros / diarios.

De esta forma, partiremos el trabajo en dos, subtrabajos que ejecutaremos desde una secuencia

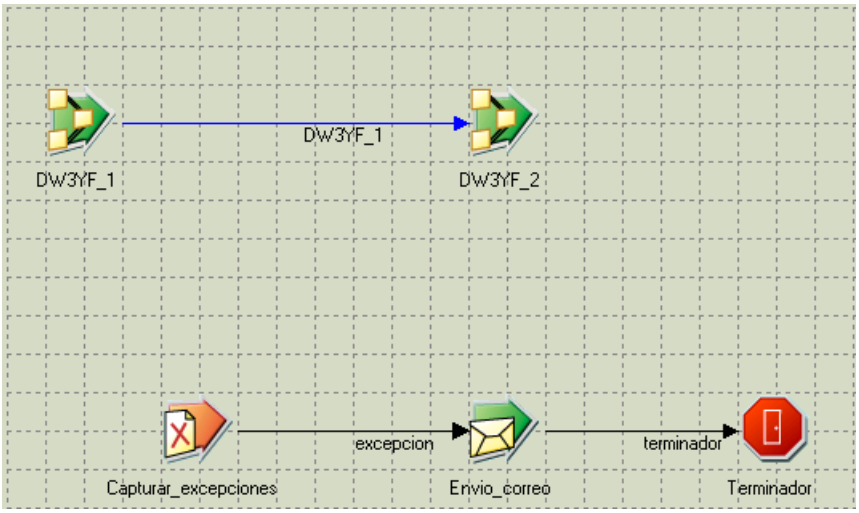

Ilustración 56 - Secuencia Totales Tickets

Las siguientes etapas se añaden a todas las secuencias para capturar excepciones en la ejecución de los trabajos y alertar mediante un envío de correo electrónico a una lista de destinatarios. La etapa "Terminador" se encarga de finalizar la secuencia actual y todas las secuencias que han llamado a ejecución a la secuencia actual.

Ilustración 57 - Etapas de control de errores (Totales Tickets)

Este sería el diseño final del primer trabajo de la secuencia:

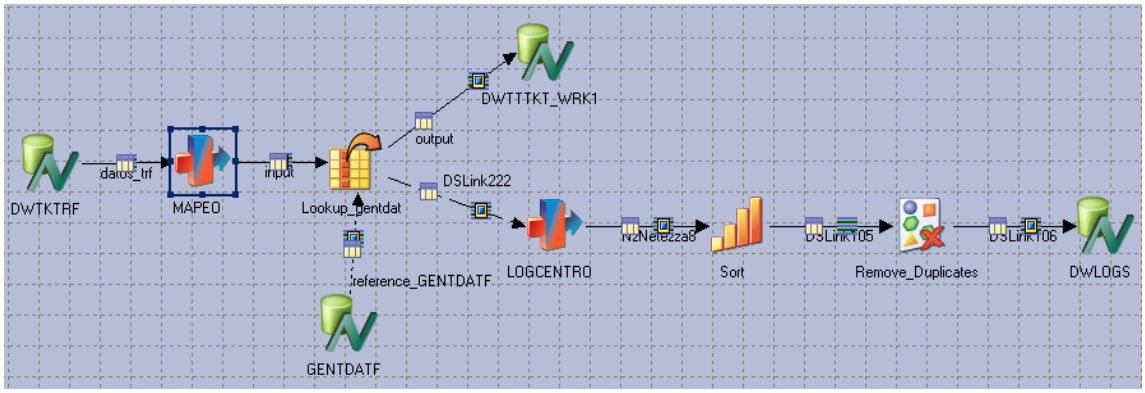

Ilustración 58 - Primera parte diseño (Totales Tickets)

Lo primero que tenemos que realizar es el acceso a los datos de origen que se encuentran ya en el sistema, en la tabla de transferencia o TRF asociada. Para ello utilizaremos una etapa de conexión a la base de datos Netezza.

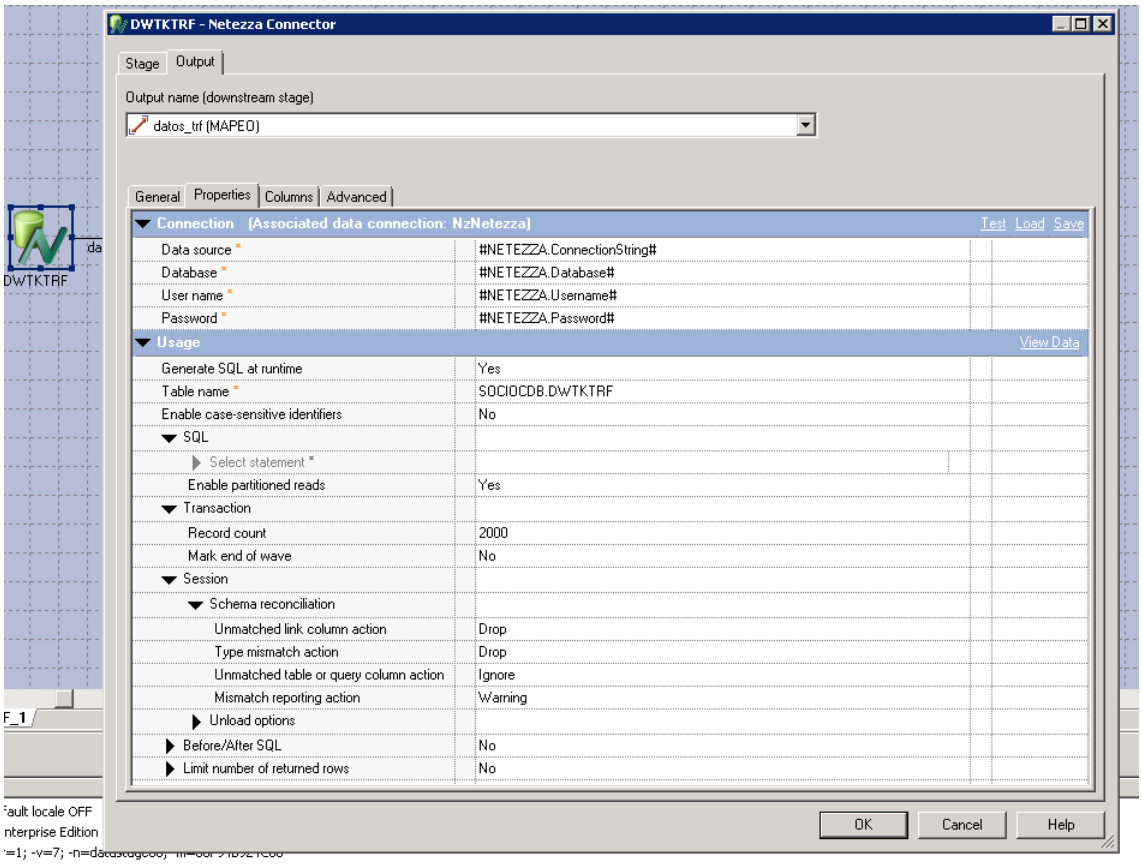

Ilustración 59 - Detalle conexión Netezza (Totales Tickets)

Configuramos los parámetros de conexión al sistema, Nombre de la base de datos, usuario y password de conexión. Definimos el nombre de esquema.tabla al que queremos acceder y definimos los nombres y tipos de datos de las columnas de origen.

DWTKTRF - Netezza Connector

Stage | Output

Output name (downstream stage)

datos_trf (MAPEO)

General | Properties | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Data element	
1	CENTR2_TXT	☐	Char		4		No		Numero de Centro 2 txt
2	FECHA_TXT	☐	Char		6		No		Fecha Movimiento txt
3	HORA_TXT	☐	Char		4		No		Hora Movimiento txt
4	NUMTIC_TXT	☐	Char		6		No		Numero de Ticket txt
5	TIFACT	☐	Char		1		No		Tipo de Factura
6	SIGNO_TIC	☐	Char		1		No		Signo ticket
7	IMPORTE1_T	☐	Char		11		No		Importe 1 txt
8	TIP_TARJ	☐	Char		1		No		Tipo tarjeta
9	SIGNO_IMTA	☐	Char		1		No		Signo impte tarjeta
10	IMPTARJ_T	☐	Char		11		No		Importe tarjeta txt
11	NUMTAR22	☐	Char		22		No		Numero de Tarjeta 22
12	SIGNO_DTO	☐	Char		1		No		Signo dto
13	IMPDESC_T	☐	Char		11		No		Importe descuento txt
14	DESCU_TXT	☐	Char		4		No		Porcentaje descuento txt
15	TARJSCTXT	☐	Char		13		No		Tarjeta fidelizacion txt
16	CAJA_TXT	☐	Char		3		No		Caja txt
17	CAJERA_TXT	☐	Char		6		No		Cajera txt
18	SINMON	☐	Char		1		No		Moneda
19	SIGNO_CUO	☐	Char		1		No		Signo cuota
20	IMPCUO_T	☐	Char		11		No		Importe cuota txt
21	SIGNO_CUP	☐	Char		1		No		Signo cupon
22	IMPCUP_T	☐	Char		11		No		Importe cupon txt
23	SIGNO_INV	☐	Char		1		No		Signo inversion
24	IMPINV_T	☐	Char		11		No		Importe inversion txt
25	DW3ZA	☐	Char		8		No		Campo Relleno 8
		☐							

Ilustración 60 - Definición de columnas, datos de origen (Totales Tickets)

Una vez que podemos leer los datos de origen nos encargaremos de realizar el mapeo de nombres de campos entre las tablas de origen y destino, así como las transformaciones de campos. Para ello utilizaremos la etapa llamada Transformer.

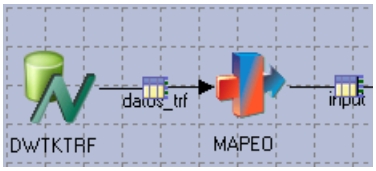

Ilustración 61 - Transfomer mapeo de campos (Totales Tickets)

Utilizando las tablas generadas del análisis del programa realizaremos los mapeos de nombres de columnas así como las transformaciones de campos.

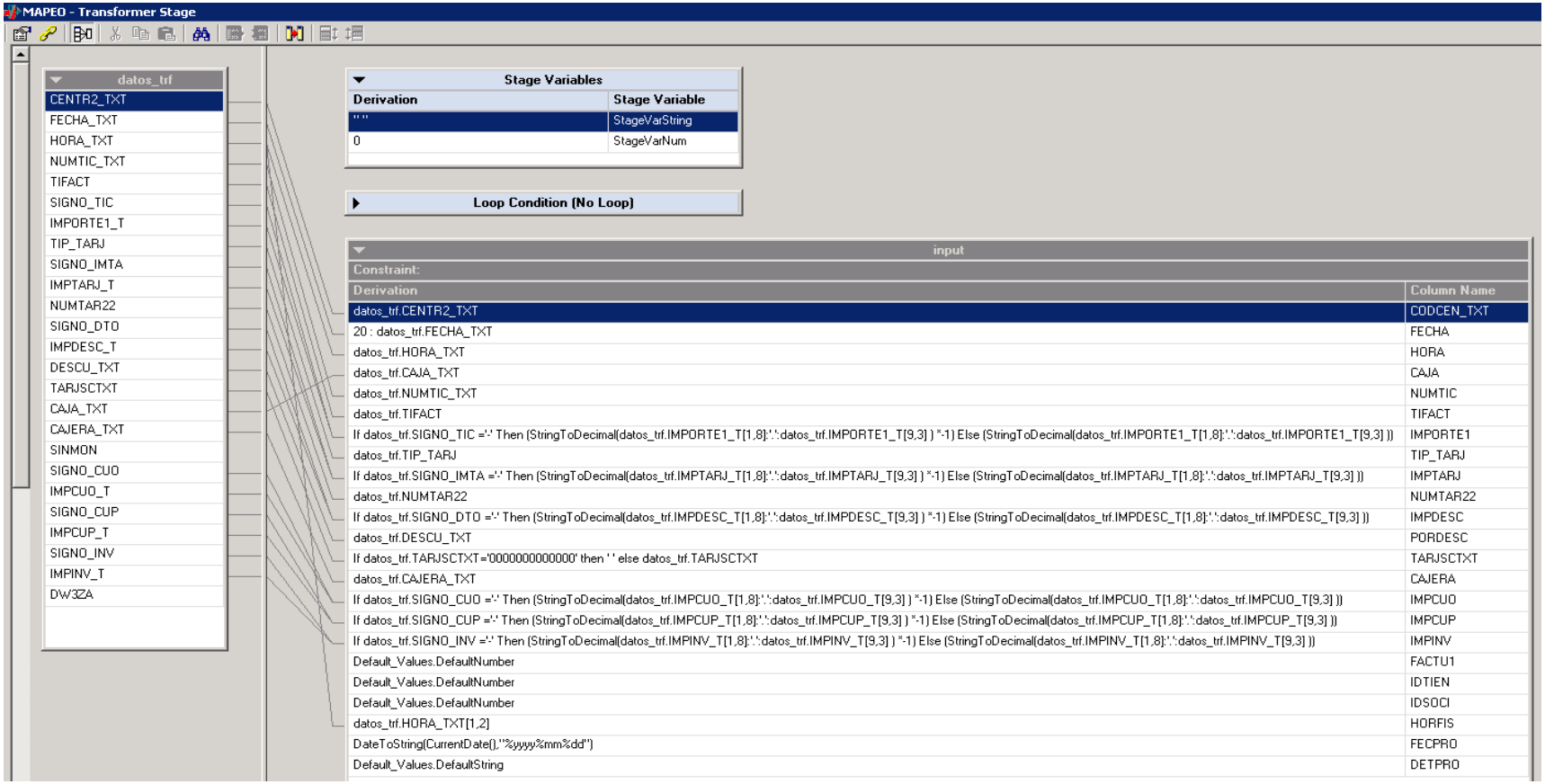

Ilustración 62 - Detalle transformaciones (Totales Tickets)

El siguiente paso es acceder a la tabla GENTDATF con GENTDATF.CODCNT=DWTKTRF.CENTR2_TXT y obtener GENTDATF.CODCLI y GENTDATF.CODCEN.

Para realizar esta operación utilizaremos una etapa de DataStage llamada LOOKUP que nos proporciona las siguientes características:

LOOKUP
Optimizado para manejar un volumen de datos "pequeño", lo que entre en la memoria ram del servidor
1 flujo de datos de entrada 1 flujo de datos de salida 1 flujo de datos de rechazados (los que no satisfacen la condición de búsqueda) N flujos de datos de referencia (para realizar las búsquedas)
Pude ejecutar sólo dos tipos de JOIN: Inner Join y Left Outer Join
No requiere que los datos estén ordenados

Tabla 8 - Etapa LOOKUP, DataStage

Los registros que no satisfagan la condición de búsqueda los insertaremos en una tabla de log de errores. De los que sí que existan en la tabla de búsqueda, extraeremos los campos requeridos y guardaremos los datos que tenemos hasta el momento en una tabla temporal en Netezza. Como podemos tener varios registros con el mismo CENTR2_TXT que no tengan un CODCNT asociado en la tabla de búsqueda, ordenamos y quitamos duplicados para insertar un único registro en la tabla de logs de errores.

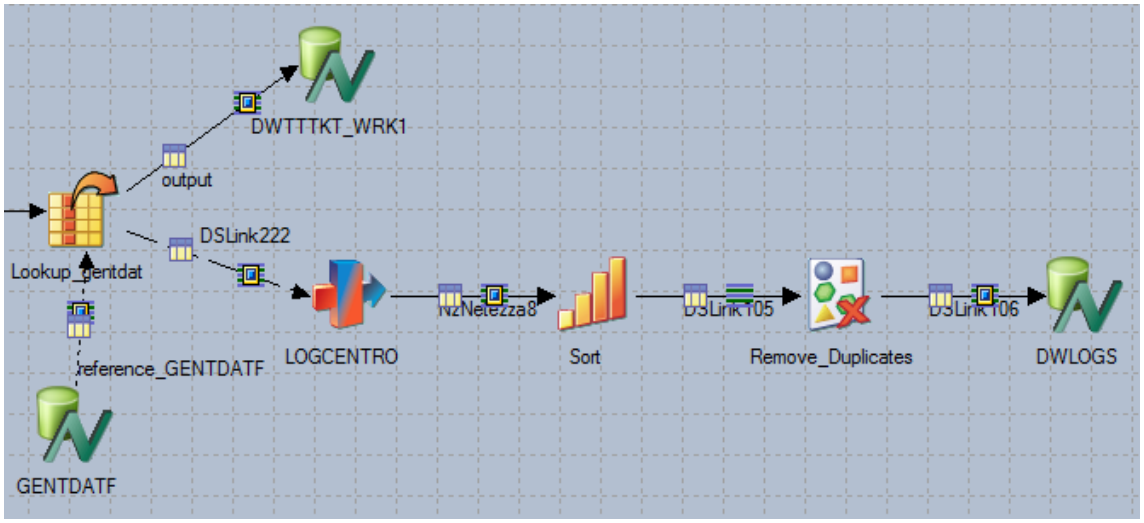

Ilustración 63 - Lookup GENTDATF (Movimiento Tickets)

El diseño de la segunda parte del trabajo es un poco más complejo, procederemos a explicarlo paso a paso.

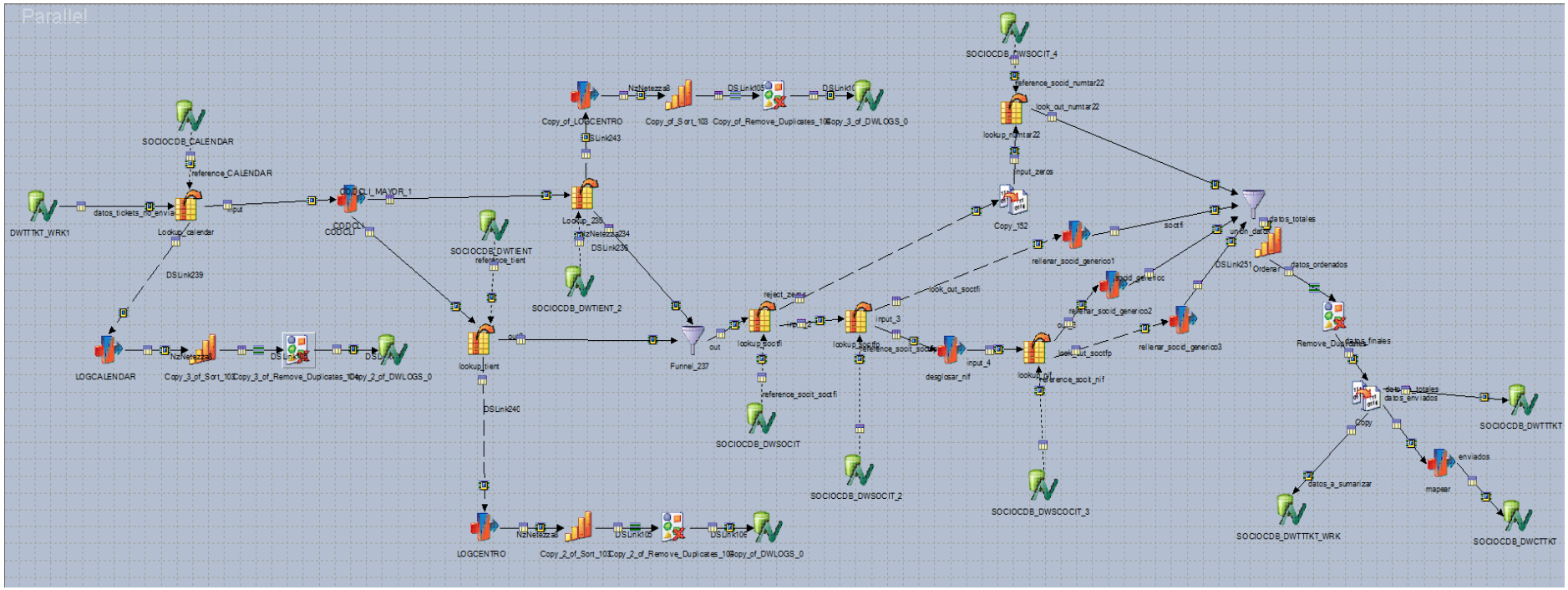

Ilustración 64 - Segunda parte diseño (Totales Tickets)

Como habíamos comentado con anterioridad, es necesario que se realice una comprobación de los datos que queremos integrar frente a la tabla que contiene el control de tickets ya procesados e integrado en el sistema, de esta forma nos aseguramos que si por algún problema externo nos llegan los datos por duplicado no insertemos la información duplicada en el sistema.

Este control se realiza al comienzo de la segunda parte del trabajo, directamente en SQL aprovechando la potencia de Netezza.

Para ello utilizamos la etapa de lectura de datos de Netezza y personalizamos el SQL para que solo nos muestre los registros de la tabla temporal (que hemos alimentado en la primera parte del trabajo) que no se encuentran en la tabla de control de tickets, siguiendo las condiciones extraídas del análisis de la lógica del programa origen.

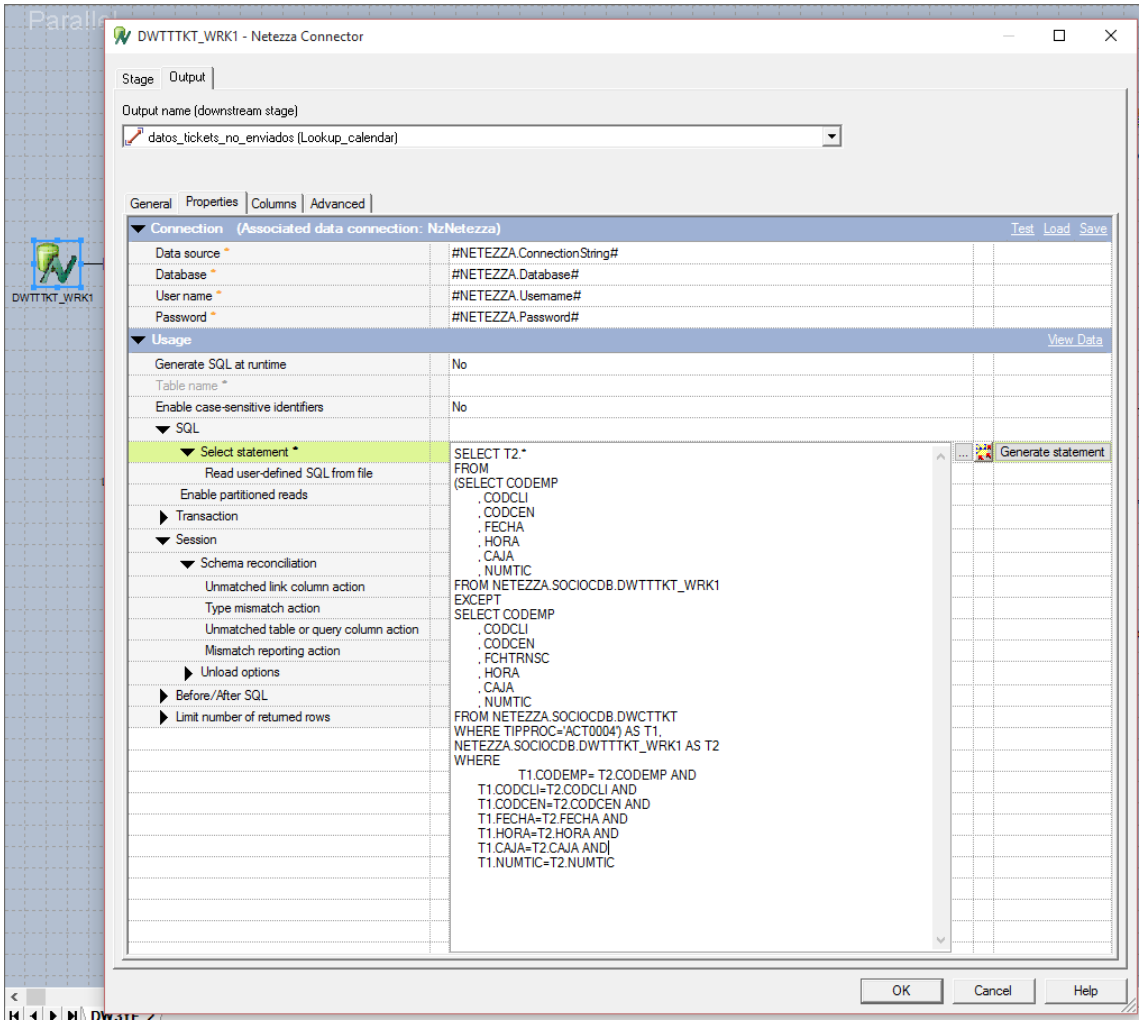

Ilustración 65 - Except Join (Totales Tickets)

Una vez que ya tenemos seleccionados los registros que cumplen en control de tickets, procedemos ir satisfaciendo el resto de búsquedas de datos en tablas foráneas que todavía tenemos pendiente.

Comenzamos con la extracción de los datos relativos a fechas:

Acceder a la tabla CALENDAR con FECHA = FECHA_TXT y recuperar ANOFIS, MESFIS, SEMFIS y DIAFIS.

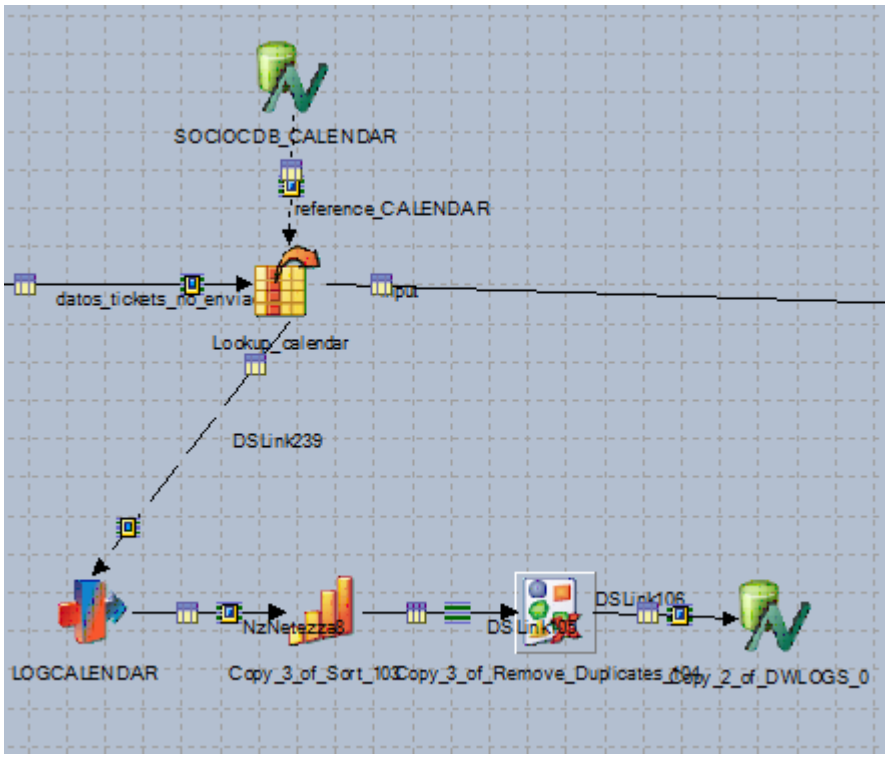

Ilustración 66 - Lookup CALENDAR I (Totales Tickets)

Los registros cuya fecha no sea correcta se envían a un fichero de log de errores.

Detalle del LOOKUP a la tabla CALENDAR

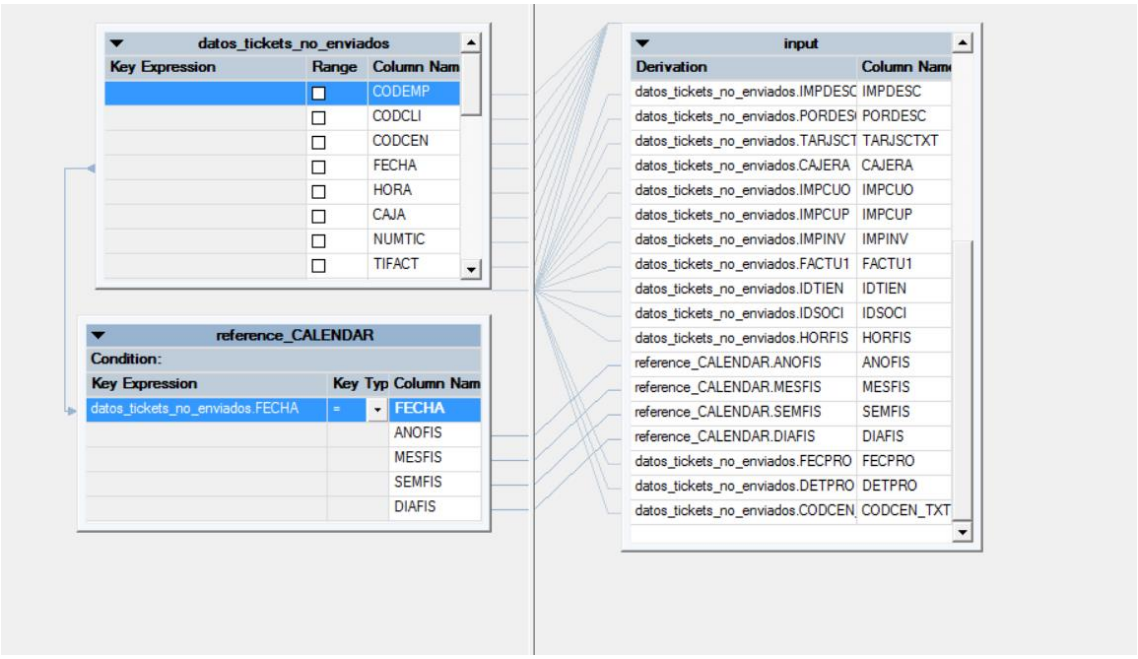

Ilustración 67 - Lookup CALENDAR II (Totales Tickets)

Seguimos con la siguiente búsqueda:

Si CODCLI>0

Acceder a DWTIENT con TIECEN = CENTR2_TXT y recuperar IDTIEN
sino
Acceder a DWTIENT con TIECEN = CODCEN y recuperar IDTIEN
finsi;

Utilizamos una etapa TRANSFORMER para dividir los datos entre los que cumplen CODCLI>0 y los que no, ya que tenemos que hacer búsquedas a la misma tabla, pero igualando campos diferente, el resultado final queremos que sea la unión de las dos ramas de datos, por lo que utilizaremos un Funnel o embudo para unificar los flujos de datos de las dos ramas lógicas.

Los datos que no disponen de IDTIEN, son rechazados en los lookups e insertados en una tabla de logs de errores.

Detalle del diseño:

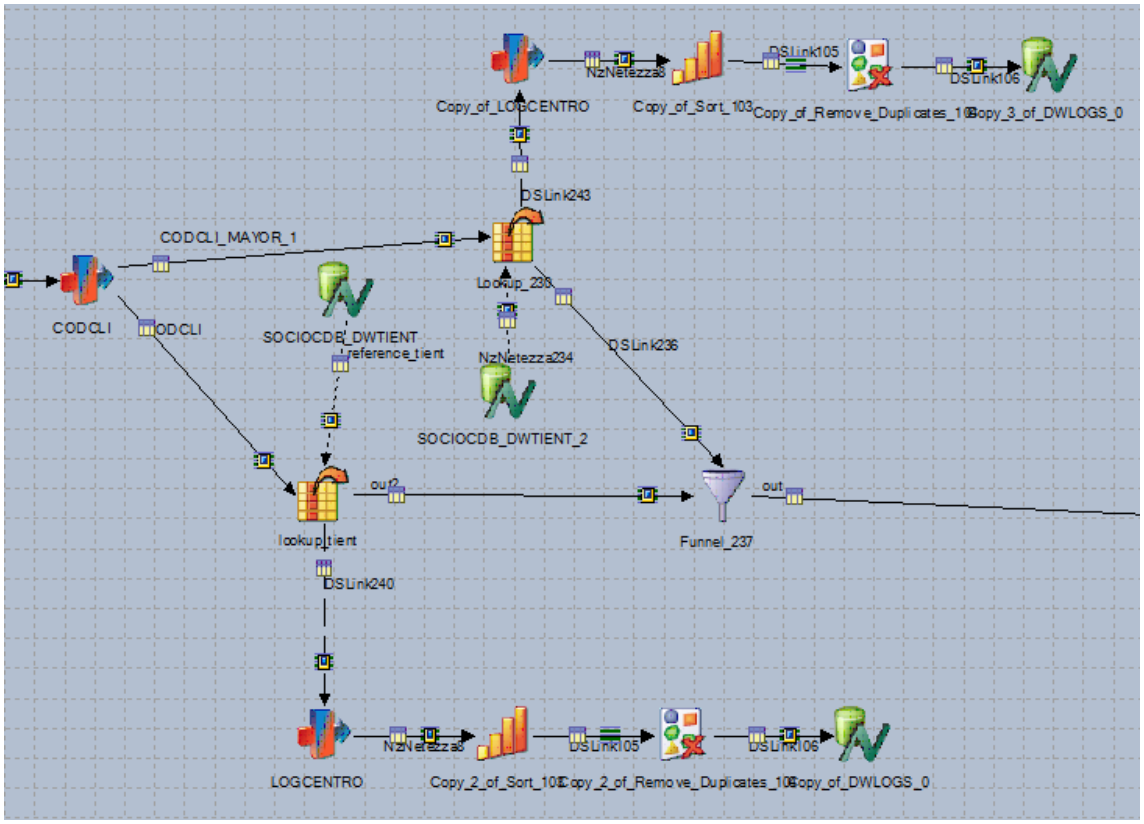

Ilustración 68 - Lookup TIENDAS (Totales Tickets)

Detalle del Transformer CODCLI

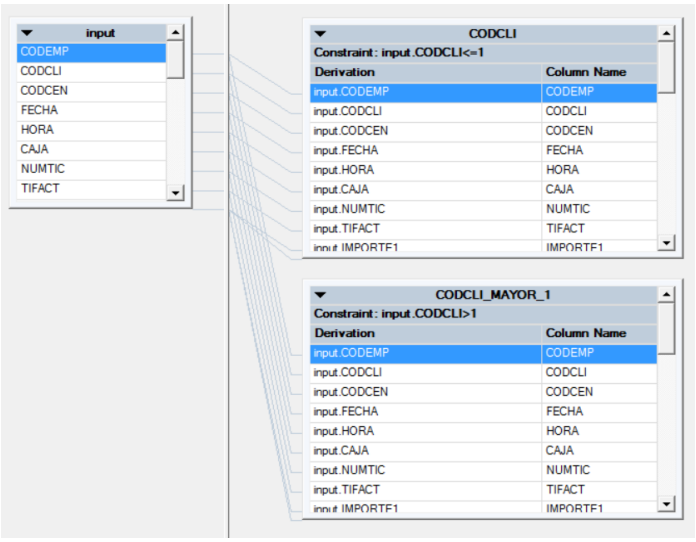

Ilustración 69 - Detalle Transformer CODCLI (Totales Tickets)

Detalle de los dos lookups a la misma tabla, pero con claves diferentes

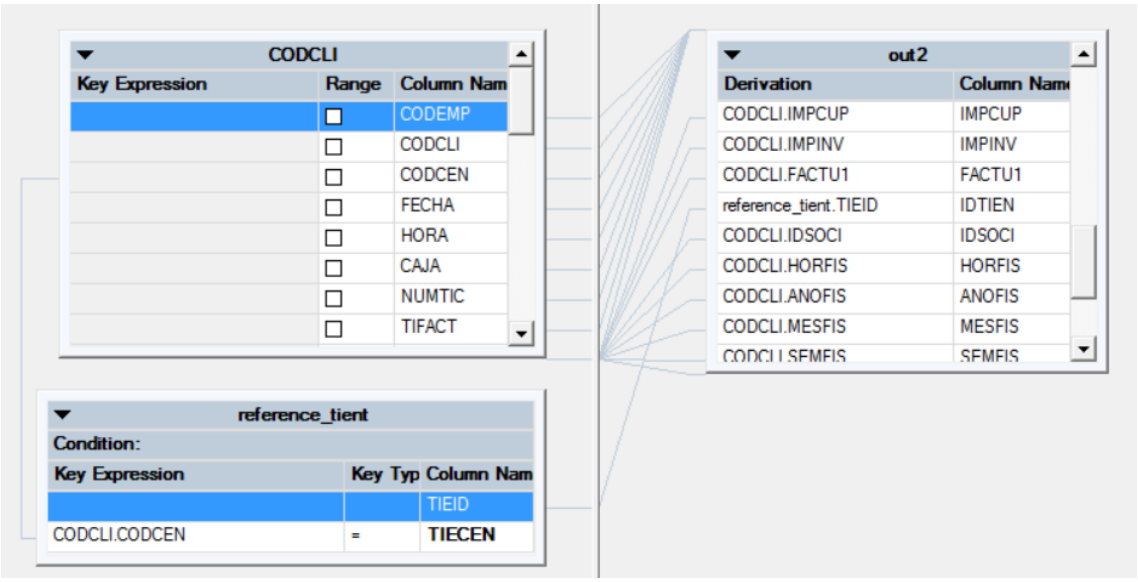

Ilustración 70 - Detalle Lookup TIENDAS I (Totales Tickets)

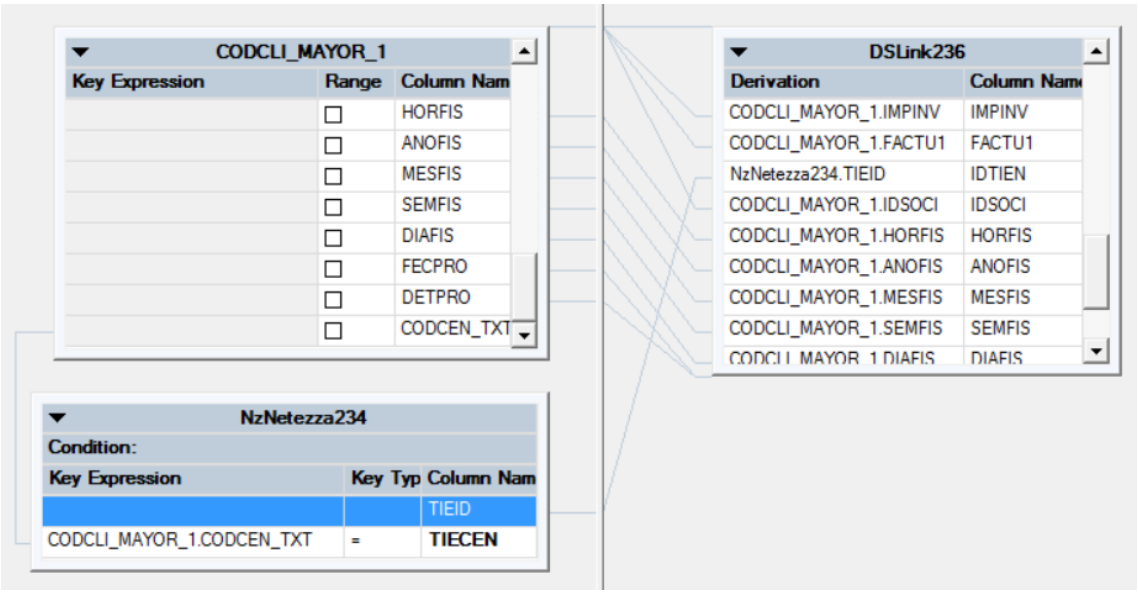

Ilustración 71 - Detalle Lookup TIENDAS II (Totales Tickets)

La última de las condiciones de búsqueda es la más compleja ya que nos presenta varios escenarios posibles a la hora de realizar las búsquedas de los datos requeridos.

IDSOCI = 0
Si TARJSCTXT <> ' ' --> **Tarjeta Fidelización**
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTFI = TARJSCTXT
Si IDSOCI = 0 --> **Tarjeta Fidelización Provisional**
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTFP = TARJSCTXT
Si IDSOCI = 0 --> **NIF**
TPJ = Primer caracter de TARJSCTXT si es una letra
IFI = letra del NIF
CPF = número NIF (9 caracteres a partrir de la 1ª ó 2ª posición de TARJSCTXT)
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTPJ = TPJ AND
SOCIFI = IFI AND
SOCCPF = PF
(Si aunque hayamos encontrado el socio, éste no tiene tarjeta fidelización,
entonces ponemos id genérico)
Si dwsocit.soctfi =' ' AND dwsocit.soctpf=' '
entonces: IDSOCI = 9999999999
Si IDSOCI = 0 --> **ID Genérico**
IDSOCI = 9999999999
Sino
Si NUMT22 <> ' ' --> **Tarjeta Pago**
IDSOCI = SELECT SOCID FROM DWSOCIT WHERE SOCTPA = NUMT22
TARJSCTXT= SOCTFI
FinSi
FinSi

En este caso, aprovecharemos la opción del LOOKUP de los rechazados para poder dividir los registros en las condiciones que necesitamos.

El diseño global de esta parte es el siguiente:

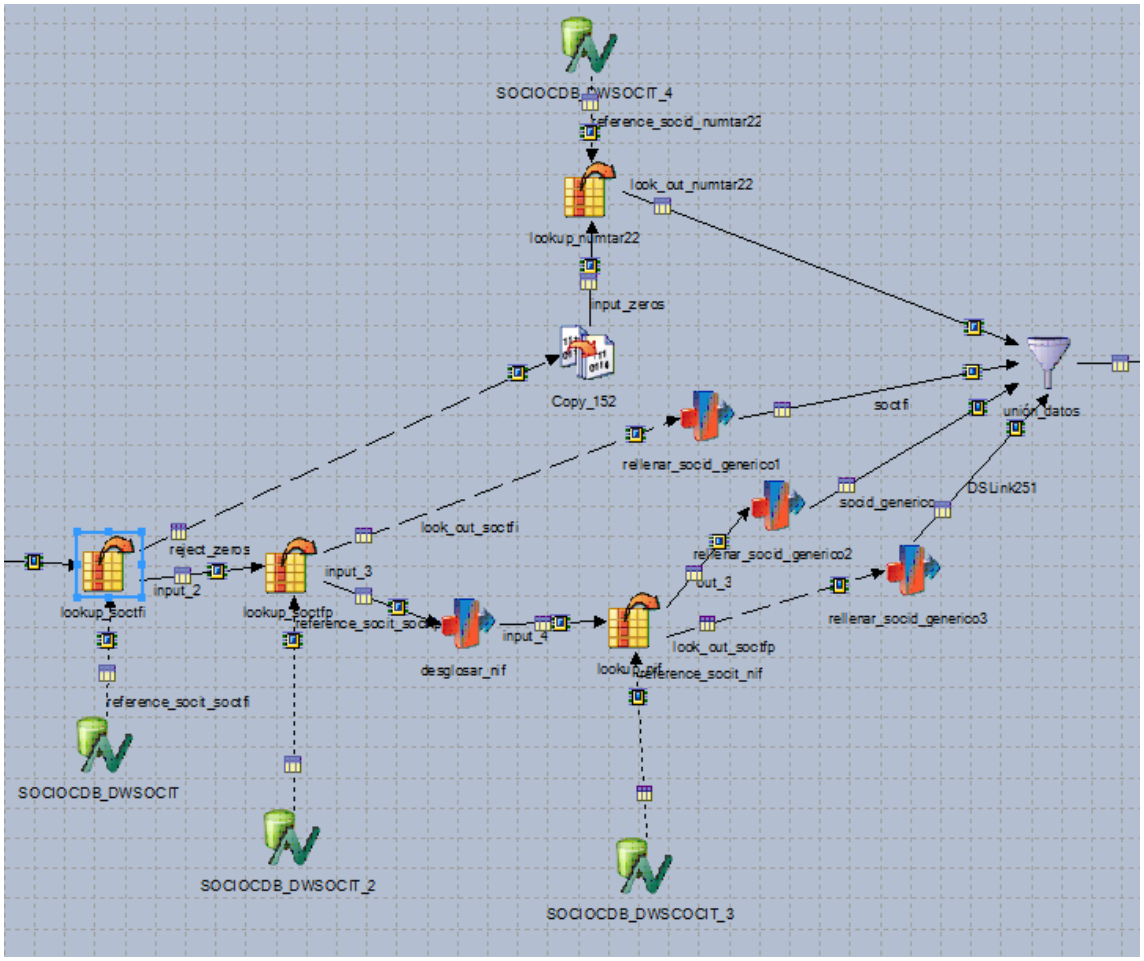

Ilustración 72 - Lookup SOCIOS (Totales Tickets)

Con el primero de los Lookup conseguimos dividir los registros entre los que tienen TARJSCTXT <> ' ' y los que no.

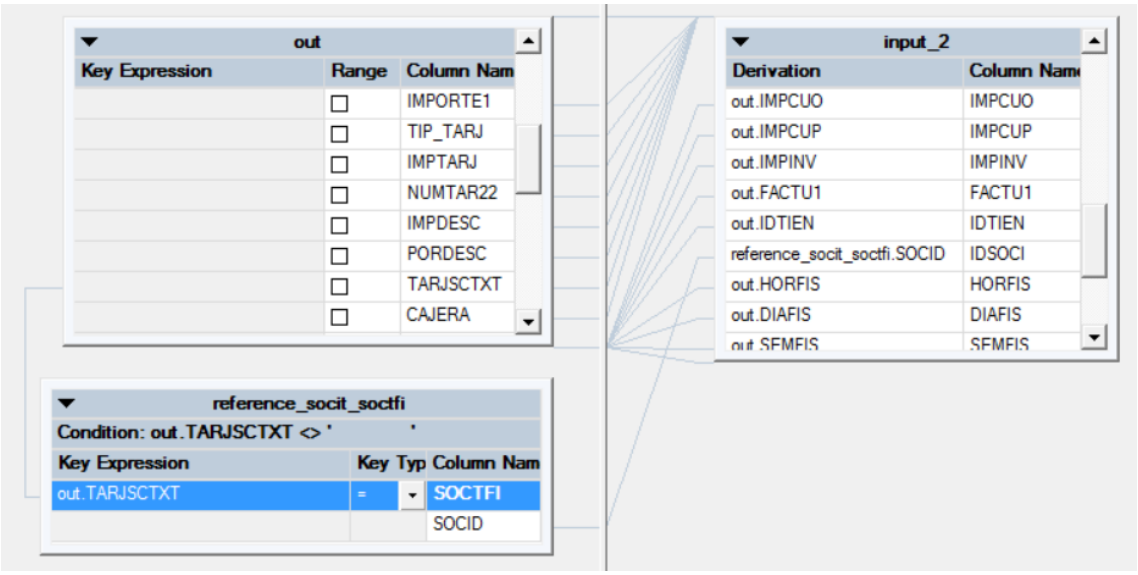

Ilustración 73 - Lookup01 SOCIOS (Totales Tickets)

Los rechazados, directamente van por una rama del flujo de datos a comprobar la condición: Si NUMT22 <> ' ', los que satisfacen esta condición se unen en un embudo con el resto de datos.

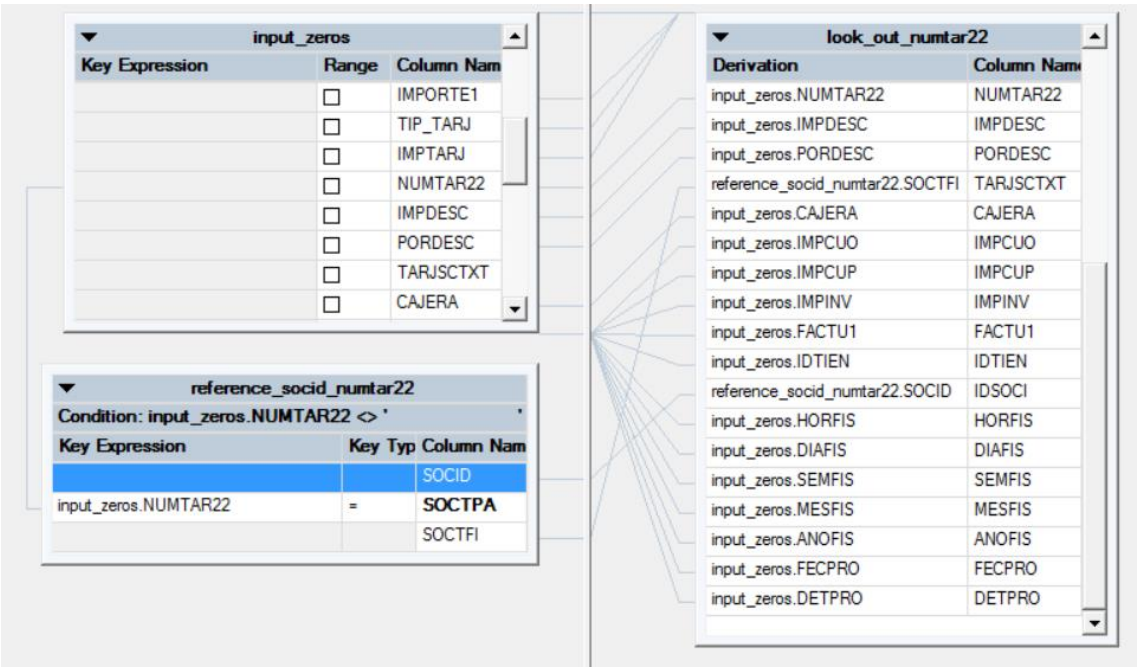

Ilustración 74 - Lookup02 SOCIOS (Totales Tickets)

Los que cumplen la condición entran en el siguiente nivel de condiciones en los que tenemos que obtener el ID del socio que realiza la compra, a partir de las posibilidades de que aparezca una tarjeta provisional o el nif en el ticket.

Detalle Lookup de los registros que tienen tarjeta e IDSOCI=0

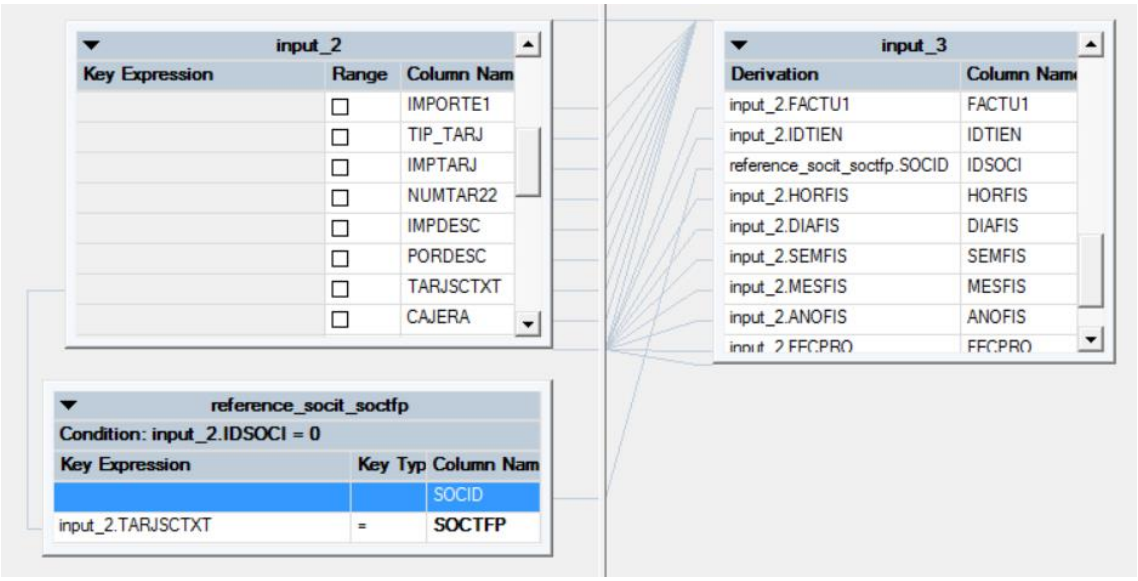

Ilustración 75 - Lookup03 SOCIOS (Totales Tickets)

A los rechazados, los que tienen tarjeta e IDSOCI <> 0 se unen en el final con el resto de flujos de datos.

Por últimos nos queda obtener los socios que tienen tarjeta pero no ID de socio, utilizando el campo NIF para realizar la búsqueda

Detalle desglose del NIF:

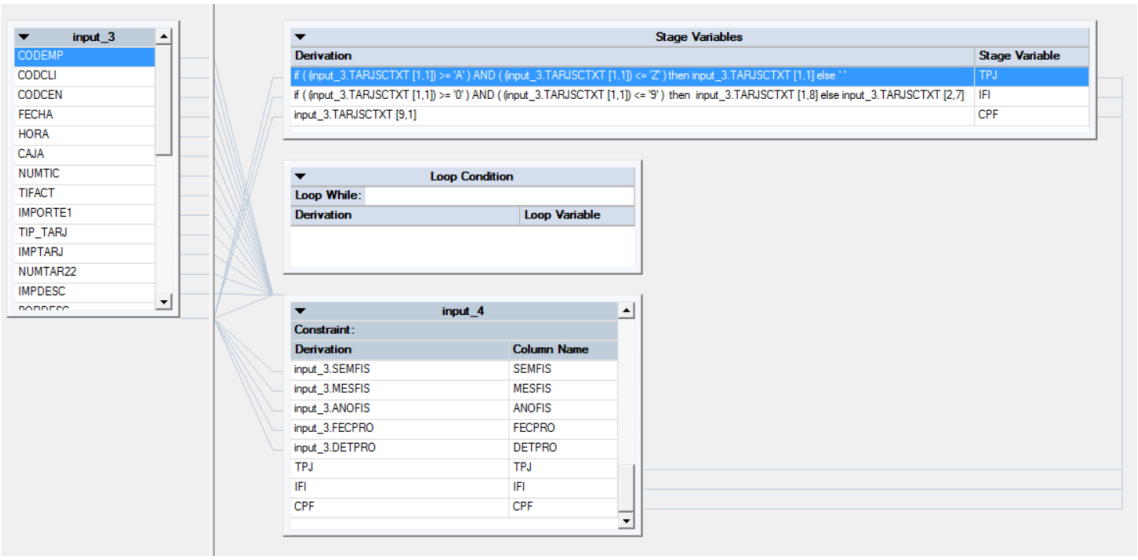

Ilustración 76 - Transformer desglose NIF SOCIOS (Totales Tickets)

Detalle Lookup socio por NIF (Desglosado en tres campos)

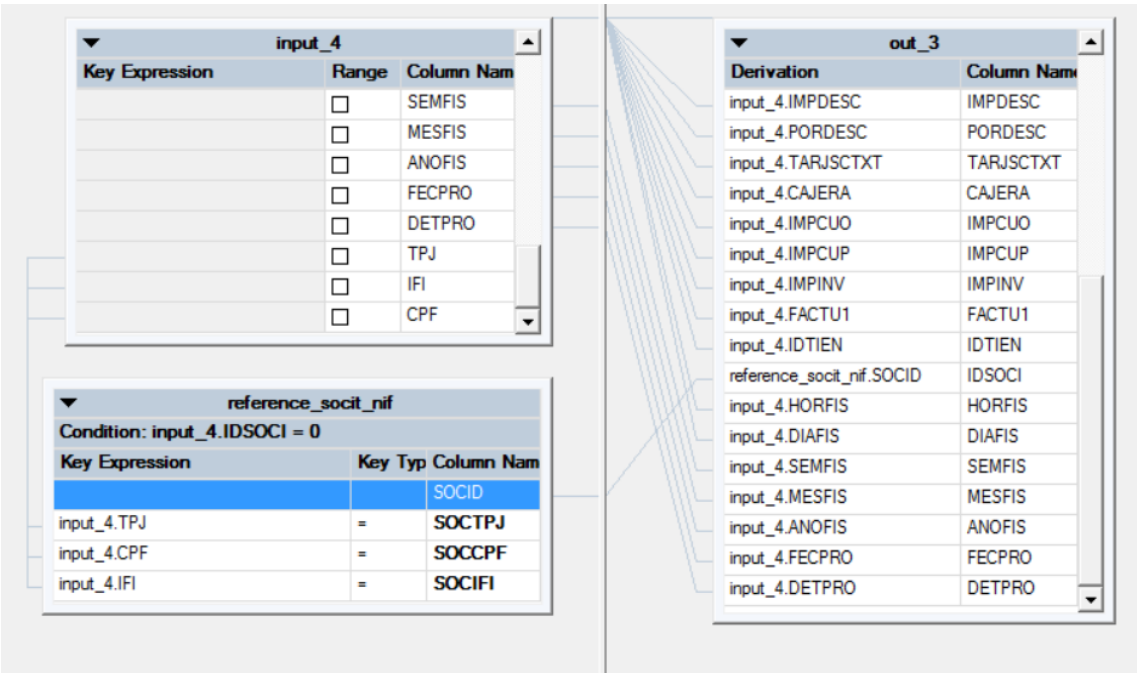

Ilustración 77 - Lookup04 SOCIOS (Totales Tickets)

Una vez que hemos conseguido abarcar todas las posibilidades de la obtención del IDSOCI unimos los flujos de datos utilizando un Funnel o embudo y procedemos a la parte final del trabajo.
En esta parte lo primero que vamos a hacer es ordenar y quitar duplicados con la clave de la tabla destino final para asegurarnos de que no insertamos ningún ticket que pueda estar repetido.

Como se comenta al inicio del análisis del programa, tenemos que insertar los datos en la tabla de tickets final DWTTTKT, pero también tenemos que insertar los datos en la tabla de control de tickets DWCTTKT y enviar los datos a una tabla temporal que se utilizará en futuros desarrollos para generar los agregados de tickets.

Detalle de las etapas necesarias para realizar las tareas anteriores:

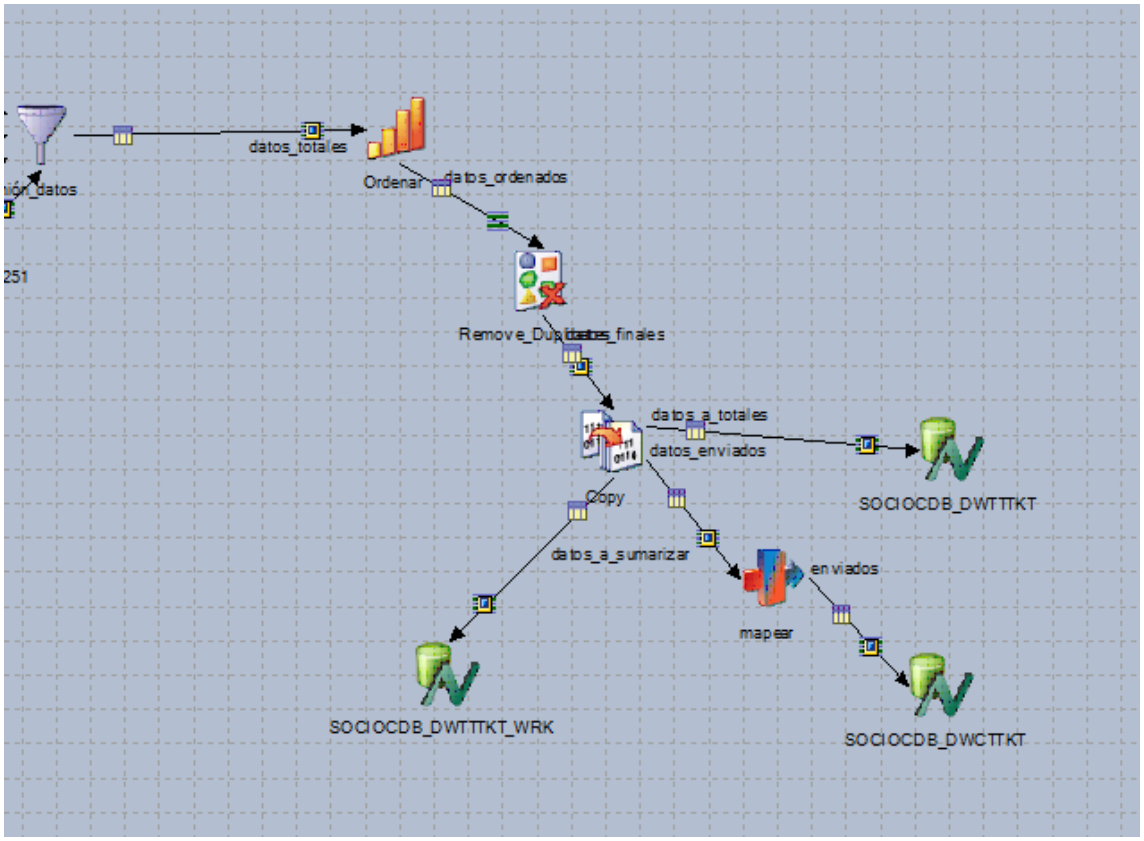

Ilustración 78 - Escritura de Datos en tablas destino (Totales Tickets)

4 CONCLUSIONES

Tal y como se describe en el punto de la memoria 1.2 OBJETIVOS, el principal objetivo que nos planteamos al comienzo del proyecto, fue mostrar la resolución de un problema real que tenía una empresa del sector de la distribución alimentaria relacionado con la optimización de la gestión del dato que estaba realizando en un almacén de datos.
Este problema abarcaba muchas áreas de trabajo dentro de la empresa, ya que afectaba a los departamentos de sistemas de la información, desarrollo y negocio.

De las dos soluciones propuestas podemos concluir que la elegida, adquisición de una solución Data Warehouse y transformación de los procesos actuales de integración de datos utilizando una herramienta ETL, apuesta por el futuro. Sin duda es la que requiere de una inversión más alta, y la que tiene un mayor riesgo, pero también es la que permite construir una base sólida sobre la que abordar futuros proyectos que de otra forma serían imposibles de realizar.

Esta solución confía en un fabricante líder del mercado como es IBM y está basada en una solución hardware como es IBM PureData (Netezza) y una herramienta ETL, InfoSphere DataStage.

La implantación de este software y hardware no solo soluciona el problema inicial planteado, sino que provoca un cambio en el funcionamiento interno de los departamentos de sistemas de la información y de desarrollo. Este cambio es costoso y llevará su tiempo, pero permite poder abordar nuevos proyectos que antes no se podían realizar o bien por la imposibilidad tecnológica o bien por el pésimo rendimiento que se iba a obtener.

De estas premisas podemos concluir que la solución elegida aun siendo arriesgada va a suponer una mejora sustancial en la gestión del dato de la empresa, lo que va a provocar una mejor gestión del negocio y un mejor posicionamiento en el sector.

Tras el análisis del problema desarrollado en el punto 2.4 ANALISIS Y PROPUESTA DE SOLUCIÓN DE PROBLEMÁTICA EN UNA EMPRESA DEL SECTOR DE DISTRIBUCIÓN ALIMENTARIA, pudimos observar que uno de los principales inconvenientes que nos encontrábamos era a nivel de rendimiento del sistema.

Al finalizar la implantación de la solución descrita en punto 3 SOLUCIÓN DATAWAREHOUSE + ETL, EN CLIENTE DEL SECTOR DISTRIBUCION ALIMENTARIA, se ha conseguido reducir los tiempos en los procesos de carga de datos, de las 24 horas del proceso en el sistema viejo a unas 2 horas en la plataforma actual. Esto ha permitido que actualmente se puedan integrar los datos 4 veces al día lo que proporciona una frescura del dato que los responsables de su explotación han agradecido.

Adicionalmente a la reducción del tiempo de integración de los datos, se han optimizado los rendimientos de los informes y consultas que se realizan sobre ellos, obteniendo mejoras de tiempo de ejecución de informes que tardaban 24 minutos aproximadamente a 10 segundos.

Esta mejora en el rendimiento de las consultas está provocando que ahora mismo el cuello de botella del circuito sea la herramienta de inteligencia de negocio, por lo que ya se está planteando una mejora tecnológica para adecuarse al nuevo entorno.

Del estudio del arte necesario para abordar el problema planteado, se marcó como objetivo mostrar una visión de un concepto, "Big Data", que en el marco tecnológico actual del dato, está "de moda" y fue el primer planteamiento que se analizó en el problema concreto descrito en el proyecto.

Tras el análisis del estado del arte podemos concluir que la revolución tecnología que está arrastrando el concepto de "Big Data" es real y puede satisfacer necesidades que hasta este momento ni siquiera nos habíamos planteado.

Sin embargo, si analizamos los proyectos que se están abordando a nivel nacional sobre esta materia podemos ver que la penetración de la misma no es tan alta como podríamos extraer si analizamos el entorno:

- Fabricantes lanzando nuevas tecnologías que cada vez tienen un tiempo de uso menor, con el consiguiente costo de formación de los recursos para su manejo.
- Bombardeo de publicidad en medios especializados.
- Demanda masiva de empleos relacionados con la gestión del dato.

Esta circunstancia es debida a que los retornos de inversión de proyectos de big data todavía no están siendo rentables para el "core" de las empresas españolas, el costo de la puesta en marcha de un proyecto de estas características es muy alto, ya no por el componente tecnológico si no por la ausencia de profesionales cualificados en la materia ya que la base de big data se ha construido sobre plataformas "open source", sin un apoyo de grandes fabricantes.

Este entorno esta comenzado a cambiar, los grandes fabricantes han visto un negocio rentable y están desplegando una batería de productos adaptando las tecnologías open source a sus sistemas, aportando el valor y la confianza de su marca.

Visualizando el conjunto global, podemos concluir que en un futuro a medio plazo los proyectos "Big Data" reales van a crecer y empresas de sectores que ahora mismo no pueden abordar esos proyectos van a ver el beneficio de realizarlos.

A raíz del análisis del problema que tenía una empresa del sector de distribución alimentaria pudimos observar que, aunque algunas empresas quieran plantearse nuevos proyectos relacionados con las nuevas posibilidades que nos proporciona el concepto "Big Data" los problemas reales del día a día imposibilitan poder realizarlos.
Para poder abordar estos proyectos necesitamos disponer de una base, tecnológica y de metodología de trabajo adecuada, moderna y capaz de gestionar el volumen, la velocidad y la variedad de los datos que vamos a manejar.

4.1 Conclusiones personales

En el plano formativo la realización de este proyecto me ha permitido explorar un área de conocimiento que tenía sin uso desde mi paso por la universidad. He podido aplicar lo aprendido sobre diseño de bases de datos y lenguaje SQL, adicionalmente me ha permitido descubrir nuevas tecnologías y herramientas relacionadas con la materia.

A nivel personal, ha supuesto un gran desafío ya que debido a mi pronta entrada en el mundo laboral y a otras circunstancias diversas, no realice el proyecto al terminar las asignaturas.

Fui demorando su escritura en el tiempo hasta llegar a un punto en el que incluso me llegué a plantear que no lo iba a poder realizar.

Independientemente de la calificación que obtenga, estoy orgulloso de, finalmente, haberlo terminado.

4.2 Trabajos futuros

El ejemplo descrito en este documento es solo una pequeña parte del trabajo necesario para migrar el entorno existente a la nueva plataforma implantada. El proyecto todavía está en marcha y tendrá un tiempo de implantación aproximado de 19 meses de trabajo para un equipo dedicado de 4 personas.

Una vez finalizado se podrán abordar nuevas mejoras, ya planteadas y analizadas en muchos casos.

Una de las posibles mejoras es reducir el tiempo de tratamiento de la información aprovechando que la lógica ya está migrada a la nueva plataforma, para ello se optimizará el proceso de carga de datos desde el origen hasta el Data Warehouse para poder abordar una cadencia de datos de aproximadamente cada 5 minutos.

Otra área que se pretende abordar es la implantación de modelos de análisis predictivos de datos aprovechando la potencia de rendimiento que nos proporciona el nuevo sistema, para ello se generaran modelos utilizando herramientas como IBM SPSS que nos proporcionaran análisis predictivos sobre cualquier inquietud que se nos ocurra. Podemos abordar modelos de fidelización de cliente, senda del abandono, segmentación de cliente, análisis del carro de la compra... las posibilidades son infinitas.

Adicionalmente al análisis predictivo de datos, con esta plataforma sí que podemos abordar proyectos de integración de datos de fuentes heterogéneas, de gran volumen y a una gran velocidad, en otras palabras “Big Data”.

Como ya se ha comentado durante las conclusiones, actualmente la mejora en el rendimiento global del sistema está provocando problemas en la herramienta de explotación de los datos. En esta área habría que realizar una propuesta de mejora tecnológica para que sea capaz de abordar los requisitos que a partir de ahora se le va a exigir.

Por último, creo que sería necesario, realizar un profundo análisis del diseño del “Data Warehouse” que hemos migrado del sistema antiguo a la nueva plataforma. Durante la ejecución de este proyecto nos hemos centrado en realizar la adaptación del sistema actual a la nueva plataforma de ejecución transformando los procesos de integración de los datos. A raíz del análisis del diseño de la base de datos hemos podido observar que no sigue los estándares de diseño de los Data Warehouses, no dispone de un esquema claramente definido, no hay una tabla de hecho rodeada de dimensiones, no cumple la mayoría de los requisitos de un almacén de datos.

Inicialmente se creó un hibrido entre un sistema transaccional y un Data Warehouse, pero ahora que se dispone de una plataforma puramente enfocada a ser un almacén de datos, sería necesario modificar el diseño mejorándolo aprovechando la experiencia adquirida durante los años de uso del sistema antiguo.

5 PRESUPUESTO

5.1 Introducción

A continuación se presentará una estimación de los costes de ejecución del proyecto. Para ello se mostrará el coste divido por las diferentes fases en las que se ha realizado, así como el tiempo empleado en cada una de ellas.

Se calcularán los costes del personal requerido para desarrollar estas tareas y los costes del material empleado.

Se considerará una jornada laboral de 4 horas al día.

5.2 Fases del proyecto

Fase de Análisis

Durante esta fase se ha realizado un estudio del estado del arte relacionado con la gestión de datos mediantes tecnologías de la información, así como la búsqueda de propuestas de soluciones Data Warehouse y ETL y el análisis del problema existente en una empresa del sector de distribución alimentaria

Para la realización de esta fase se han utilizado los siguientes recursos:

Recurso	*Tiempo empleado (días)*
Ordenador Lenovo E531	10
Ingeniero Técnico	10

Tabla 9 - Recursos y tiempo empleado, fase análisis

Fase de Desarrollo

Durante la fase de desarrollo se ha realizado la prueba de concepto para determinar la solución ganadora. Se ha implantado la solución HW+SW tanto del Data Warehouse como de la ETL. Adicionalmente se ha realizado una transformación de un trabajo del sistema antiguo a la nueva plataforma

Para la realización de esta fase se han utilizado los siguientes recursos:

Recurso	*Tiempo empleado (días)*
Ordenador Lenovo E531	44
IBM PureData System For Analytics	44
8286-42A IBM Power S824	35
IBM Data Architect	1
IBM InfoSphere DataStage: Designer	35
Aginity Workbench for Netezza	15
PuTTY	1
Director de Proyecto	3
Ingeniero Técnico	44

Tabla 10 - Recursos y tiempo empleado, fase desarrollo

Fase de Documentación

Durante esta fase se ha reunido toda la información generada en fases previas y se ha documentado la memoria del proyecto

Para la realización de esta fase se han utilizado los siguientes recursos:

Recurso	*Tiempo empleado (días)*
Ordenador Lenovo E531	15
Ingeniero Técnico	15
Microsoft Office 2013	15

Tabla 11 - Recursos y tiempo empleado, fase documentación

A continuación se muestra un diagrama de Gannt del proyecto:

Nombre de tarea	Duración	Comienzo	Fin
Proyecto	**69 días**	**mié 01/07/15**	**lun 05/10/15**
Incio de proyecto	0 días		
Fase Analisis	**10 días**	**mié 01/07/15**	**mar 14/07/15**
Analisis del estado del arte	3 días	mié 01/07/15	vie 03/07/15
Busqueda propuestas Data Warehouse	2 días	lun 06/07/15	mar 07/07/15
Busqueda propuestas ETL	2 días	mié 08/07/15	jue 09/07/15
Analisis problemática de cliente	3 días	vie 10/07/15	mar 14/07/15
Fase Desarrollo	**44 días**	**mié 15/07/15**	**lun 14/09/15**
Prueba de Concepto	7 días	mié 15/07/15	jue 23/07/15
Arquitectura Data Warehouse	5 días	vie 24/07/15	jue 30/07/15
Arquitectura ETL	7 días	vie 31/07/15	lun 10/08/15
Analisis y diseño caso de uso	25 días	mar 11/08/15	lun 14/09/15
Fase Documentacion	**15 días**	**mar 15/09/15**	**lun 05/10/15**
Recopilacion de datos	1 día	mar 15/09/15	mar 15/09/15
Generar la memoria del proyecto	14 días	mié 16/09/15	lun 05/10/15

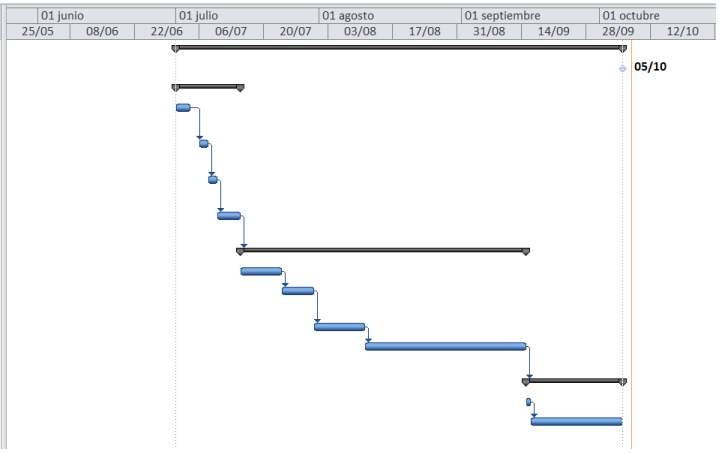

Ilustración 79 - Diagrama Gantt del proyecto

5.3 Costes

5.3.1 Coste de personal

Durante la realización del proyecto han intervenido dos personas:

RECURSO	HORAS DEDICADAS AL PROYECTO	PRECIO UNITARIO (€/h)	TOTAL PRECIO (€)
DIRECTOR DE PROYECTO	12	100	1200
INGENIERO TECNICO	276	65	17940
		TOTAL:	**19.140,00 €**

Tabla 12 - Costes de personal para la realización del proyecto

5.3.2 Coste de material y herramientas

Detalle del coste material y de herramientas empleadas durante el proyecto

DESCRIPCIÓN	COSTE (€)	% USO DEDICADO AL PROYECYO	DEDICACIÓN (Meses)	PERIODO DE DEPRECIACIÓN	COSTE IMPUTABLE
Lenovo E531	595	100	3,45	60	34,21
Office Professional 2013	399	100	0,75	36	8,31
IBM PureData System for Analytics	750000	100	2,2	48	34375,00
8286-42A IBM Power S824	120000	100	1,75	48	4375,00
IBM InfoSphere DataStage	12000	100	1,75	48	437,50
				TOTAL:	**39.230,03 €**

Tabla 13 - Costes de material y herramienta para la realización del proyecto

El presupuesto total de este proyecto asciende a la cantidad de 58.370,03€ (CINCUETA Y OCHO MIL TRESCIENTOS SETENTA EUROS CON TRES CENTIMOS)

Leganés a 05 de Octubre de 2015

El ingeniero proyectista

Fdo. UNAI FERNANDEZ RIVAS

6 GLOSARIO

ETL	*Extract Transform and Load*
IDG	*International Data Group*
IBM	*International Business Machines Corp*
MRP	*Material Requirements Planning*
SQL	*Structure Query Language*
ERP	*Enterprise Resource Planning*
BI	*Bussines Intelligence*
EB	*ExaByte*
HTML	*HyperText Markup Language*
BBVA	*Banco Bilbao Vizcaya Argentaria*
CERN	*Conseil Européen pour la Recherche Nucléaire*
NASA	*National Aeronautics and Space Administration*
AMPP	*Asymmetric Massively Parallel Processing*
TB	*TeraByte*
PCI	*Peripheral Component Interconnect*
DW	*Data Warehouse*
CPU	*Central Processing Unit*
VM	*Virtual Machine*
XML	*eXtensible Markup Language*
CDC	*Change Data Capture*
TCP	*Transmission Control Protocol*
JSON	*JavaScript Object Notation*
JDBC	*Java DataBase Connectivity*
WAS	*WebsSphere Application Server*
RPG	*Report Program Generator*
COBOL	*Common Business-Oriented Language*

PoC	*Proof Of Concept*
DDL	*Data Definition Language*
BTU	*British Thermal Unit*
SCSI	*Small Computer System Interface*
SAS	*Serial Attached SCSI*
RAID	*Redundant Array of Independent Disks*
FPGA	*Field Programmable Gate Array*
Kb	*Kilobit*
KB	*KiloByte*
HDD	*Hard Disk Drive*
SDD	*Solid State Drive*
CAPI	*Computer Assisted Programming Interface*
PCIe	*Peripheral Component Interconnect Express*
DVD	*Digital Versatile Disc*
LUN	*Logical Unit Number*
ODBC	*Open DataBase Connectivity*
SAP	*Sistemas, Aplicaciones y Productos*
DB2	*Database System Management*
FTP	*File Transfer Protocol*
OLTP	*OnLine Transaction Processing*
RAW	*Read After Write*
SOA	*Service Oriented Architecture*

7 REFERENCIAS

- [Viktor Mayer-Schönberger, Kenneth Cukier 2013] Big Data: A Revolution That Will Transform How We Live, Work, and Think World Wide Web: https://books.google.es/books?id=HpHcGAkFEjkC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- [IDG 2015] Encuesta "2015 Big Data and Analytics Survey" World Wide Web: http://www.idgenterprise.com/report/2015-big-data-and-analytics-survey
- [Relational Database 2015] Historia de la Base de datos relacional World Wide Web: http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/reldb/
- [MRP 2015] Sistemas de Planificación de las necesidades de material World Wide Web: http://www.factoryphysics.com/documents/leanWP1.pdf
- [Devlin et Murphy, IBM Systems Journal 1988] An architecture for a business and information system World Wide Web: http://9sight.com/EBIS_Devlin_&_Murphy_1988.pdf
- [D. J. Power 2007] A Brief History of Decision Support Systems World Wide Web: http://DSSResources.COM/history/dsshistory.html
- [Michael Cox et David Ellsworth 1997] Application-Controlled Demand Paging for Out-of-Core Visualization World Wide Web: http://www.nas.nasa.gov/assets/pdf/techreports/1997/nas-97-010.pdf
- [Peter Lyman et Hal R. Varian 2000] How much information? World Wide Web: http://www2.sims.berkeley.edu/research/projects/how-much-info/how-much-info.pdf
- [Davis Mendoza Paco] Logical Volume Manager World Wide Web: http://geo.gob.bo/blog/spip.php?article97
- [Tim O'Reilly 2005] What Is Web 2.0, World Wide Web: http://www.oreilly.com/pub/a//web2/archive/what-is-web-20.html
- [Mike Olson] HADOOP: Scalable, Flexible Data Storage and Analysis, World Wide Web: http://www.cloudera.com/content/dam/cloudera/Resources/PDF/Olson_IQT_Quarterly_Spring_2010.pdf
- [Gartner 2009] World Wide Web: http://www.gartner.com/newsroom/id/855612
- [McKinsey Global Institute] Big data: The next frontier for innovation, competition, and productivity, World Wide Web: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation

- [IBM Knowledge Center 2014] InfoSphere DataStage infocenter World Wide Web: http://www-01.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/com.ibm.swg.im.iis.productization.iisinfsv.overview.arch.doc/topics/wsisinst_arch_client_layer.html?lang=es
- [Inmon W 1992] Bulding The Data Warehouse, QED Techinical Publising Group World Wide Web: https://books.google.es/books?id=9T6Oe6AujzUC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- [Carlos Fernández, Dataprix] ¿Qué es un Data Warehouse? World Wide Web: http://www.dataprix.com/qu-es-un-data-warehouse
- [Sinnexus] Datawarehouse World Wide Web: http://www.sinnexus.com/business_intelligence/datawarehouse.aspx
- [Oracle Exadata] World Wide Web: http://www.oracle.com/lad/engineered-systems/exadata/database-machine-x5-2/features/index.html
- [Teradata DataWarehouse] Teradata DataWarehouse data sheet World Wide Web: http://www.teradata.com.es/Resources/Datasheets/Teradata-Data-Warehouse-Appliance-2800/?LangType=1034&LangSelect=true
- [Marketing Director] Teradata Data Warehouse Appliance World Wide Web: http://www.marketingdirecto.com/actualidad/digital/teradata-data-warehouse-appliance-2800-satisface-las-demandas-analiticas-mas-exigentes/
- [CIO PERU] Teradata Data Warehouse Appliance World Wide Web: http://cioperu.pe/articulo/12861/teradata-anuncia-nuevo-appliance-para-data-warehouse/
- [Gravitar] Appliances para Data Warehouse y Business Intelligence World Wide Web: http://gravitar.biz/bi/appliance-data-warehouse-business-intelligence/
- [DotHill] AssuredSAN World Wide Web: http://globenewswire.com/news-release/2015/04/20/725934/10129582/en/Teradata-Integrates-Dot-Hill-AssuredSAN-Ultra48-Into-New-Data-Warehouse-Appliance-2800.html
- [PowerData] Procesos ETL World Wide Web: http://max.beta.upcnet.es/23/upcnet-cs/ca/proves-k-l/procediment-backup-sap/powerdata_-_procesos_etl.pdf
- [Roberto Espinosa, DataPrix 2010] Herramientas ETL World Wide Web: http://www.dataprix.com/blogs/respinosamilla/herramientas-etl-que-son-para-que-valen-productos-mas-conocidos-etl-s-open-sour#
- [Juan Garcés 2013] Introducción a INFORMATICA POWERCENTER World Wide Web: http://www.jgarces.info/introduccion-a-informatica-powercenter/
- [Code Jobs 2013] ¿Qué es Informatica PowerCenter? World Wide Web: https://www.codejobs.biz/es/blog/2013/10/24/que-es-informatica-powercenter

- [Oracle 2015] Introduction to Oracle Warehouse Builder World Wide Web: http://docs.oracle.com/cd/B28359_01/owb.111/b31278/concept_overview.htm
- [Oracle 2015] Understanding the Basic Concepts World Wide Web: http://docs.oracle.com/cd/B28359_01/owb.111/b31278/concept_basics.htm
- [InfoSphere DataState Designer] World Wide Web: https://www-01.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/com.ibm.swg.im.iis.ds.design.doc/topics/cont_designing_ds_and_qs.html
- [IBM InfoSphere Data Architect] World Wide Web: http://www-03.ibm.com/software/products/es/ibminfodataarch
- [Aginity Workbench Netezza] World Wide Web: http://www.aginity.com/workbench/netezza/
- [PuTTY] World Wide Web: https://es.wikipedia.org/wiki/PuTTY